SHORT TEXT CLASSIFICATION OF CLINICAL QUESTIONS

by

SHUBHAM JINDAL

(Under the Direction of PROF. KHALED M RASHEED)

ABSTRACT

A clinical question is a question that a health care provider may have, during a patient encounter in a clinical setting. This thesis describes the design and implementation of a text classification system to automatically classify a clinical question into various categories based on a hierarchical taxonomy designed by researchers. The system implements natural language processing and machine learning based techniques to automatically classify a clinical question. This question classification system would be integrated into an Evidence-Based Point-of-Care clinical decision support system providing concise, practical, readily accessible information from various resources to facilitate fast and accurate decision making. The system would analyze clinical questions on both the syntactic and semantic level (extraction of noun phrases and nouns, identifying them like disease/syndrome using ontology). This study presents an initial prototype and the long term goal of the project is to classify all types of clinical questions with good results.

INDEX WORDS: Natural Language Processing, Machine Learning, Short-text, Clinical Question, Text Classification

SHORT TEXT CLASSIFICATION OF CLINICAL QUESTIONS

by

SHUBHAM JINDAL

Bachelor of Technology, Bharati Vidyapeeth University, Pune, India, 2012

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2016

© 2016

Shubham Jindal

All Rights Reserved

SHORT TEXT CLASSIFICATION OF CLINICAL QUESTIONS

by

SHUBHAM JINDAL

Major Professor:	Khaled M Rasheed
Committee:	Walter D Potter
	Mark H Ebell

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2016

DEDICATION

I would like to dedicate this Master's Thesis to my family, my extended family and friends for their unconditional support. And to my 'little' Jaan.

ACKNOWLEDGEMENTS

I would like to thank my advisor and mentor, Dr. Khaled Rasheed, for teaching us the machine learning course and for all the support and guidance I have received during my Masters study at the university. His kindness and encouragement were especially helpful when I was feeling low and really needed the support, without which this thesis project work would not have come to conclusion. He has always shown enthusiasm whenever I wanted to pursue my own research interest and it has been an honor to do this graduate work under his direction.

I would also like to thank my thesis committee members, Dr. Don Potter and Dr. Mark Ebell. Dr. Potter has taught me how to push myself one extra step, and the value of working to the best of my abilities till the very end. I am very thankful to him for always guiding me when I felt lost and showing me the right direction. I am very grateful to Dr. Ebell and was lucky to have access to his invaluable knowledge in the medical field. This research was partly supported by a Graduate Research Assistantship under Dr. Ebell and I would like to thank him for guiding me through this project. He has always shown patience while answering my questions and I am indebted to him for giving me some very valuable professional advice that helped me greatly during my Summer Internship at a later time. I would especially like to thank Dr. Michael A. Covington for teaching us natural language processing techniques which enabled me to execute this project.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

OBJECTIVE

Question answering systems have to process questions in order to extract out the relevant information from them. These questions can be in any format, natural language free-text format or in a structured query format. The extracted information from the questions can then be used to search for the required information in the databases, knowledge bases, and various sources to form relevant answers.

Clinical questions are the questions that a physician may have during a patient encounter and asking clinical queries is one of the methods that physicians use to learn during their practice. Those physicians have limited time and resources available to them at this time. The accepted procedure is to practice evidence based medicine which focuses on using the best available accurate information to support those health care professionals in answering their clinical queries. There are various knowledge resources that are available to the physicians and it is found that PubMed, a searchable database of biomedical articles, provided by the National Library of Medicine, is the most up-to date source for published journals in this field.

The clinical questions that are asked by physicians and other healthcare providers can range from being very simple to very complex during the point of care. The physicians can generally afford up to two minutes to search for an answer while it can take up to thirty minutes for a health care professional to find an answer (Ely 2005).

Question answering systems in the clinical domain have only recently begun to make progress driven by the Text REtrieval Conference (TREC) as the clinical questions have complex restricted domain-specific terminologies. The success of a clinical question answering system would depend on how fast it can provide accurate and concise information to the physicians in response to their questions. Not only does the question answering system have to make use of the evidence-based paradigm in order to find accurate information but also it must give concise answers in a fast, timely manner to the physician during the point of care. The question answering system can often fail due to information overload.

It is thus clear that classifying clinical questions into appropriate categories and then, using this classification to give concise, specific and accurate answers can greatly help us. The classification of questions into various categories can help an information retrieval system to a large extent. Classifying the questions into appropriate categories can greatly reduce the search space for an information retrieval system as to where to look for information during the search.

This thesis describes the design and implementation of a text classification system to automatically classify a clinical question into various categories based on a hierarchical taxonomy designed by researchers. This taxonomy follows a five level hierarchical arrangement containing a total of 64 generic categories of questions. The text classification system would implement different techniques based on natural language processing and machine learning in order to categorize any question into these categories.

The proposed text classification system would analyze clinical questions on both the syntactic level (part-of-speech tagging, parsing) and the semantic level (using an

ontology to find a semantic interpretation) to extract the proper features from the clinical questions. Then these extracted features from the questions from the previous step would be used as feature values for different categories of the questions to implement machine learning techniques on them.

It is also proposed to develop this text classification system using a modular approach so that enhancements could be easily added in the future. Another unique component of this research study is that the programming language for the proposed text classification system is Prolog, thus applying artificial intelligence technologies to build applications that could be used in the real world medical domain. However this thesis project should be considered a preliminary work providing an initial evaluation for the proposed system and developing an initial prototype for the future research teams on this project to work on.

OVERVIEW

In the rest of this thesis, Chapter 2 goes through the background and the related research, the taxonomy of the clinical questions developed by the researchers, the data collection of the clinical questions, and the challenges posed by the classification task. Then Chapter 3 describes the lexical pattern matching module implemented and the various steps within it. It then describes the results and discussion. Chapter 4 describes the feature extraction phase of the project using natural language processing. Chapter 5 then gives an overview of the medical database UMLS and the MetaMap program. Chapter 6 describes the experiments and the results with machine learning using the two approaches. Finally, Chapter 7 presents a discussion of the project and then chapter 8 discusses the conclusion and the future research directions.

CHAPTER 2

RELEVANT BACKGROUND

RELATED WORK

The biomedical domain is a restricted domain and has particular characteristics which pose challenges for any application to be developed for the biomedical/clinical domain which is very different from a biological domain as the settings are very specific to medical and clinical related fields.

Evidence based medicine is the paradigm in the biomedical world that emphasizes on using the most recent research articles and knowledge sources to search for any evidence related to any medical concept. Thus, evidence based medicine focuses on using the best available accurate information to help support the health care professionals.

A valuable resource for any interested researcher in the biomedical domain is MEDLINE, a bibliographic database and collection of abstracts provided by the National Library of Medicine (NLM). It is found that PubMed, a database of biomedical articles, also provided by the NLM, is the most up-to date source for the published journals in this field.

Clinical questions are the questions that a physician may have during a patient encounter and asking clinical questions and finding answers to these questions are among the methods that physicians use to learn during their practice. Lucchiari et al. (2012) discusses the various errors that the doctors make during the diagnostic process and provides a conceptual schema for the expert systems. Thus, a clinical decision support

system integrated with a clinical question answering system could help the physicians by providing accurate and concise information in response to their questions. A useful approach could be classifying the questions into appropriate categories to greatly reduce the search space for an information retrieval system as to where to look for information during the search. It is therefore proposed that a question classification system needs to be developed in the question answering system to first classify the queried clinical question into an appropriate category and then, use this categorized question to feed into the information retrieval system to give concise, specific and accurate answers in a timely manner.

The lack of a large corpus to carry out further research had also hindered the much needed progress. However, MEDLINE is a valuable resource for any interested researcher in the biomedical domain. The NLM also maintains a list of terminologies like concepts, diseases and drugs in the Medical Subject Headings (MeSH). De Leo et al. (2006) reported in their study that most physicians prefer a targeted site rather than utilizing a search engine like Google. The largest database that could be accessed by the healthcare professionals is PubMed which also facilitates finding relevant documents in the database based on the user's query and matching it with MeSH.

The most popular format for question formulation used in the recent research on the clinical question answering domain in the Evidence based medicine is the PICO format (Niu et al. 2003). PICO stands for Problem, Intervention, Comparison, and Outcome. However, it is not always convenient for healthcare professionals to formulate their questions in the PICO format. Ely et.al (2005) reported the obstacles to finding

answers that healthcare professionals face and thus also developed a hierarchical taxonomy to categorize the clinical questions.

Recent conferences and challenges like Clinical NLP (Natural Language Processing) Challenge have further fueled the application of natural language processing techniques to the clinical field. Clinical NLP Challenge 2007 (Pestian et. al. 2007, Sasaki 2007, Suominen et. al. 2008) engaged the researchers to assign ICD-9-CM (International Classification of Diseases, Ninth Revision, Clinical Modification) codes to free-text radiology reports, which share the similarity with the clinical questions of being in free-text natural language in the clinical domain and of very short lengths. The 2010 i2b2 (Informatics for Integrating Biology and the Bedside) NLP shared task involved the challenge of finding concepts, assertions and relations in clinical text.

Pestian et al. (2012) discussed the various natural language processing techniques that could be applied and the lexical resources needed specifically for the clinical text in a clinical setting. Johnson et al. (1989) provides a method using sublanguage analysis to prepare a controlled medical vocabulary and how the sublanguage could also be used to analyze the clinical questions. Lakiotaki et al. (2013) provided a method to classify medical documents for use between experts and novice users based on clinical queries using UMLS. Jonnalagadda et al. (2013) provided an information extraction method from Medline for the clinical questions related to the treatment for depression and Alzheimer's disease.

Many clinical research applications can be realized using machine learning based approach. Tong et al. (2002) devised new algorithm with active machine learning using Support Vector Machines for text classification thus depending less on the labeled trained

instances. Berger et. al. (1996) showed the effectiveness of machine learning using the maximum entropy method for natural language processing. Agrawal et al. (2010) developed biomedical machine learning-based NLP system called NegScope and HedgeScope that can detect negation and hedging using Conditional Random Fields (CRFs) in the clinical notes. Lancet, a machine learning system, can extract medicines names and other related information like dosage, frequency, duration from clinical discharge summaries.

Mayo cTAKES (Savova et. al. 2010), an open source NLP system for information extraction from clinical free-text uses both rule based approach and machine learning techniques in their various modules like sentence boundary detector, named entity recognition, etc. Garla et al. (2013) developed a biomedical word sense disambiguation system using machine learning, integrated with cTAKES and showed that it improves clinical document classification. There have been huge recent developments in question answering system by the development of IBM Watson. As explained by Lally et al. (2012), it uses complex parsing and semantic rules to identify features, classify it and detects critical elements of the question. However the system is proprietary and is developed with a large commercial space and budget.

ARCHITECTURE OF THE PROPOSED CLINICAL DECISION SUPPORT SYSTEM

A question answering system has to process questions in order to extract out the relevant information from them. The extracted information can then be used to search for the required information in the databases, knowledge bases and various sources to form relevant answers.

A clinical decision support system integrated with a clinical question answering system could help the physicians by providing accurate and concise information to the physicians in response to their questions. A useful approach could be classifying the questions into appropriate categories to greatly reduce the search space for an information retrieval system as to where to look for information during the search.

After the questions are classified into the very precise types of categories, the questions could then be compared with the templates that are associated with these categories. These templates could provide us with the necessary information and features that are extracted by the template matching process. For instance, this would tell us that the question is a diagnostic question asking about the cause of a symptom.

The information retrieval engine then has to only look for the "cause" of that particular "symptom", instead of the usual keyword based search that usually causes information overload. Also the information retrieval would only look for this information in the recent articles and domain specific knowledge sources to give us concise answers.

The architecture of the proposed system can be described to be consisting of the modules shown in Figure 1:

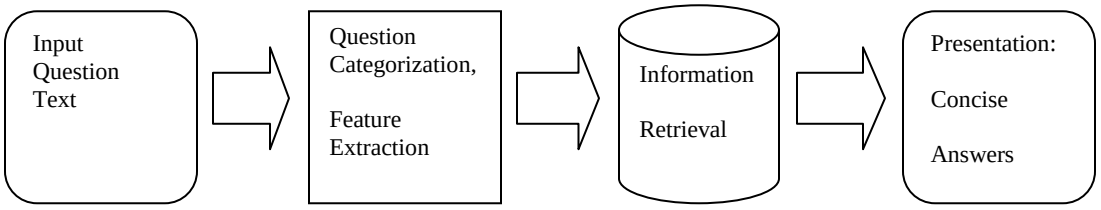

Figure 1: Architecture of Clinical Decision Support System

Not only must the question answering system make use of Evidence based paradigm in order to find accurate information but also it must give concise answers in a fast, timely manner to the physicians during the point of care.

A TAXONOMY OF CLINICAL QUESTIONS

A taxonomy of 64 generic categories of clinical questions has been developed by Ely and his colleagues (Ely 2000) using 1396 questions as their dataset which is tabulated in Table 5 in Appendix B. These categories follow a four-level hierarchical arrangement such that at the topmost level of the hierarchy the questions could be broadly classified into five categories - diagnosis, treatment, management, epidemiology, non clinical (a possible sixth category could also be defined at this level as 'unclassified'). Within each of these five topmost-level categories (primary level), the question can be further categorized into a secondary-level specific categories, which could then be further categorized into tertiary-level specific categories, which could again be further categorized into the final quaternary-level categories.

The number of categories within the secondary, tertiary or quaternary levels is not uniform across the five topmost primary levels. For instance, although there are 8 categories at the secondary level within the topmost 'diagnosis' primary level & only 3 categories at the secondary level within the topmost 'treatment' primary level, there are a total of 18 question categories (cumulative of secondary, tertiary and quaternary levels) within the topmost 'diagnosis' primary level, while a total of 23 question categories (cumulative of secondary, tertiary and quaternary levels) within the topmost 'treatment' primary level.

It is also noted that the distribution of the number of questions belonging to each category, from the original dataset of 1396 questions, is not uniform. The three most frequent questions fall into these generic categories - “What is the drug of choice for condition x?” (11%), “What is the cause of symptom x?” (8%), and “What test is indicated in the situation x?” (8%). It is further found that approximately 80% of the clinical queries can be found in only 20% of the question types.

For example, a question like 'What is the optimum dose of penicillin for streptococcal sore throat in a 5-year old boy?' would be coded as ‘treatment’ (primary level), "drug prescribing" (secondary), "how to prescribe" (tertiary) and "dosage" (quaternary). So, this question would be categorized into the generic question category 2.1.1.2 as "What is the dose of drug x in situation y?" where 2 is the code for the category ‘treatment’ at the primary level; at the secondary level within the topmost-level ‘treatment’ category, 1 is the code for ‘drug prescribing’ category; at the tertiary level within the secondary ‘drug prescribing’ category, 1 is the code for ‘how to prescribe’ category; and finally at the quaternary level within the ‘how to prescribe’ category, 2 is the code for ‘dosage’ category.

Another question like 'What is the best antibiotic for a streptococcal carrier?' would be coded as "treatment" (primary level), "drug prescribing" (secondary), "drug of choice" (tertiary) and "treatment" (quaternary). So, this question would be categorized into the generic question category 2.1.2.1 as "What is the drug of choice for situation y?” It is to be noted that this question and the one stated before both have the ‘drug prescribing’ category at the secondary level, but this category at these two secondary levels are actually entirely two different categories (2 different categories among the 64

generic flat categories) as they are contained within the two different primary level categories. Similarly, in this question, 'treatment' category at the primary level is different from the 'treatment' category at the quaternary level (2 different categories among the 64 generic flat categories).

QUESTION DATA SET

The question data set used for this thesis project is a collection of 4655 clinical questions, provided by the National Library of Medicine (NLM). The questions were collected from different healthcare professionals across the U.S. healthcare providers and each of these questions has been manually categorized according to the taxonomy developed by Ely and his colleagues. Each of these questions is present in three formats - short question, general question and original question.

For instance, a question could be presented in any of these forms -

Short Question: What would cause red lesions on his heel?

General Question: What would cause painful red subcutaneous lesions on the heel of an 8 year-old boy?

Original Question: 8-year-old male with painful red bumps on his heel. Red painful subcutaneous, 3 to 4 millimeters in diameter. Nothing helpful.

The original questions were the questions recorded by the data collectors during the point of care round and represent the natural free-text form of the questions that the physicians actually ask. The short questions and the general questions are the questions that were manually reformatted by various annotators. The short and general questions are the questions that were considered by these annotators to be of real practical use for a question answering system. Hence, for the thesis project only short questions and general

questions are considered, since the original questions were considered too difficult to be representative of the questions posed to a question answering system.

CHALLENGES FOR THE PROPOSED CLINICAL QUESTION CLASSIFICATION SYSTEM

The clinical question classification system poses several challenges in order to be developed, which can be summarized as:

- A restricted domain (biomedical) text classification problem which poses challenges of being a complex domain.
- Specific terminology used which is not in use in everyday-life general questions.
- Lack of a large size corpus & questions databases.
- Small training set.
- Domain-specific three formats of questions.
- Questions in free-text complex natural language not conforming to syntactic rules.
- Questions of very short length, usually a maximum of 20 words, and so various statistical measures do not generate good results.
- A multi-label text classification.
- Large number of classes (64 types) for simple classification problem.
- Questions that may be only semantically different while being syntactically very similar. For example, consider these three questions – “Can symptom x cause disease y?”, “Can disease y cause symptom x?” and “Can drug x cause finding y?”.

The proposed question answering system would need to address these challenges using the questions based on the taxonomy developed with the described data set.

CHAPTER 3

QUESTIONS ANALYSIS AND LEXICAL PATTERN MATCHING

METHODOLOGY

In this project, the clinical questions that are to be classified automatically were first analyzed for their structure. Finding some patterns that these questions might follow, could give us an insight to what rules would be most accurate for the specific categories. This thesis research project was coded in Prolog using SWI Prolog v 6.2. Prolog is very efficient for rule-based queries and when it was developed it was originally intended to be used for natural language processing. Thus, it becomes an ideal choice for this project. With the rule based approach, the following steps are performed.

Data Set Preprocessing and Analysis

This includes preprocessing the questions data set and analyzing it to get familiar with the data set. This involves taking the following steps.

Data Collection: The questions data set was collected from the NLM repository available at ClinicalQuestions Collection (2013). The collection consisted of a total of 4655 clinical questions split over four XML-formatted files. The number of questions contained in each XML file were 1095, 1892, 1062 and 606 respectively. Every question in the files was formatted with 39 division tags as shown in Appendix A.

Data Extraction: A tool was developed to analyze the XML file structure by using the XML Schema File (XSD) that was also available with each XML file. After analyzing the structure, the data was converted from XML files to the relational database using this

tool. The tables in the database were further queried to process and convert them into Microsoft Excel File Worksheet (XLS). The code for this preprocessing tool was written in Java and MySQL, as it was noted that this code would not be used for our text classification system, rather only initially to extract the data from the original data set. Therefore it should not have any effect on this project in the future, being not written in Prolog, as it would not be used again, since all the data from the original XML files have been extracted into the portable and human readable XLS Excel file. The Excel file could also be readily converted into CSV file.

Data Sampling: The end product from the above step is a huge Excel file which consists of a table where each row is a question record, and the columns are the original division tags for each of this question record. Thus, there were 4655 rows and 39 columns. These columns contain important information about each question as it was the actual classification by a human annotator. Out of these 39 columns, three columns are used for this project. These three columns are original question, short question, and general question. The data from these three columns were sampled out using Microsoft Excel.

Data Set Analysis: On analyzing these three question types in the Microsoft Excel program, it was found that many of these questions were missing in the record. For instance, in a record, the original and short question fields are present while the field for the general question is empty, or the original and the general question are present but short question field is missing. Also, the questions in the data set were not uniformly distributed. The class type 2.1.2.1., the generic type question “What is the drug of choice for condition x?” has the highest number of instances in the data set. The data set was also analyzed to find the most frequent question types.

Sampling Training Examples: For this project, the questions dataset of 4655 clinical questions were divided into three sets as training set (60%), validation set (15%), and testing set (25%). The questions were all shuffled and the division was done at a random order in Microsoft Excel. Only the training set of 2793 questions is used and the remaining questions were separated for later stages of validation and testing.

Text Preprocessing: The text files were pre-processed to remove all double quotes (“), percentage sign (%) and other symbols or punctuation marks that are not helpful in text classification. Initially the single quote (‘) was also removed, but then it was found to be necessary for many possessive nouns and pronoun words. The questions were checked for any spelling errors using a word processor, and the errors were corrected.

Question Type Specification

The questions were manually analyzed to find a structure and to observe patterns that these questions might conform to. As a first step, it was considered to try to analyze the category of diagnostic questions (categories 1.1.1.1, 1.1.2.1, 1.1.3.1) and classify them correctly. This category was chosen as these questions were in the five top most frequently occurring questions, and also finding the accurate answers to the diagnoses questions is one of the most challenging task for medical health care professionals. Thus, it was decided that finding patterns in these questions could serve as a useful initial first step for this project and in the later stages of the project, more pattern matching rules could be further identified for all the 64 categories. These diagnostic questions can be thought of as “What cause” questions, as they generally belong to the type “What is the cause of symptom x?” The training set of the questions was analyzed for the questions of the type that seek to inquire “if finding X could be a symptom of condition Y” in various

forms as “What is the cause of symptom X?” This rule-base approach was followed by lexical and syntactic analysis as described below.

Question Reclassification

In this step, a thorough analysis of the questions in the training set was performed and it was found that many of the questions were not correctly classified as per the Ely’s taxonomy. So, many questions have to be reclassified and given correct codes. Further, the questions of our interest “What cause” were identified and it was found that these questions majorly belong to the category 1.1.1.1, 1.1.2.1, 1.1.3.1 and 1.1.4.1 where at primary level, code 1 represents ‘diagnosis’, at secondary level, code 1 represents ‘cause/interpretation of clinical finding’ and at tertiary level, codes 1, 2, 3 and 4 represents symptom, sign, test finding and unspecified findings respectively. These questions were reclassified by a medical health care professional whose domain knowledge in the medical field was necessary. One of the researchers on this project is also a distinguished health care provider and the question reclassification of the diagnostic questions was performed by him so that the previous errors could be corrected. However, during this question reclassification task, it was also found that the questions in our training set occurring in the category 1.1.4.1 were not truly representative of the diagnostic questions. And so the questions belonging to this category were removed from the diagnostic category set.

Question Analysis and Keyword Identification

In this step, various rules are identified by manually analyzing all the short and general questions in the training set that could be potentially used to classify the questions into “What cause” diagnostic type. A few rules that helped substantially are

- Presence of words ‘what’ and ‘cause’, and their morphological forms in the question and that the word ‘cause’ is to follow the word ‘what’. (E.g. what would cause coccyx pain with no history of injury?).
- Presence of words ‘can’ and ‘cause’ and their morphological forms in the question and that the word ‘cause’ is to follow the word ‘can’. (E.g. can chronic alcoholism cause diarrhea?).
- Presence of words ‘why’ and ‘does’ and their morphological forms in the question and that the word ‘does’ is to follow the word ‘why’. (E.g. why does this infant have hyponatremia?).
- Presence of words ‘differential’ and ‘diagnosis’ and their morphological forms in the question. (E.g. what is the differential diagnosis of a blue skin lesion on the abdomen?).
- Presence of words ‘distinguish’ and ‘possibility’ and their morphological forms in the question. (E.g. what are the possible causes of bone marrow suppression, and what is the best way to distinguish among the possibilities?).
- Presence of words ‘due’ and ‘to’ and their morphological forms in the question. (E.g. is the cough due to her asthma or to the angiotensin converting enzyme inhibitor?).

Tokenization

Tokenization is a procedure where the text is broken down into separate words called tokens. A token can be a contiguous series of letters, digits or special character by itself. Whitespace characters separate token. The program code is written in Prolog. The questions from the text files, generated by the word processor during the preprocessing

step, are input to the tokenization code. The questions in the text file have also been removed of their category codes assigned to them. At the end of this step, the question text is converted into a list of Prolog atoms. The detailed procedure taken for this step is explained in the next chapter. At the end of this step, the words of the text are separated and stored for further processing in our lexical pattern matching module.

Pattern matching

The Prolog atoms output from the previous step was taken as input and the keywords and the pattern matching rules that were identified during the previous analysis are implemented in this module. After inspecting the input text with the pattern matching rules stated in the component, the question text is classified as the diagnostic or non-diagnostic question. The module can also be passed a text file containing a list of questions, where each question is separated on a new line. The module counts the number of questions classified in the category of the diagnostic questions and the total number of questions and gives measure in terms of recall, specificity and precision.

RESULTS

For this module of the thesis project, the performance is evaluated on the short and general questions from the training set and the testing set for the classification of ‘what cause’ questions. The evaluation parameters used are Recall (or Sensitivity), Specificity, Precision (or Positive Predictive Value). The recall is a measure of number of positive instances correctly predicted by the system, or true positive rate. Specificity is a measure of number of negative instances correctly predicted by the system, or true negative rate. Precision is the number of correct instances divided by the total number of instances predicted by the system.

For a total number of 2793 questions in the training set and 1162 questions in the testing set, the classification was done on a binary level as diagnostic versus non-diagnostic questions and the results are summarized below in the Table 1 and Table 2:

Table 1: Results on the training set for binary classification

	Recall	**Specificity**	**Precision**
Short Questions	60.71%	95.49%	62.54%
General Questions	76.49%	93.22%	58.48%

Table 2: Results on the testing set for binary classification

	Recall	**Specificity**	**Precision**
Short Questions	56.83%	96.97%	71.82%
General Questions	77.04%	95.20%	68.42%

DISCUSSION

The pattern matching rules seem to perform very well on the diagnostic questions with respect to specificity, but not so much on recall. High specificity shows how these pattern rules are so specific to the diagnostic questions. Usually it's a trade-off between recall and specificity, rules that address to increase the recall tends to decrease the specificity. The general questions had much higher recall than short questions, while overall short questions had only slightly better specificity than general questions and better precision than general questions.

The process of finding patterns for each category of questions requires a lot of time and is quite laborious. While working to find patterns in the diagnostic question versus non-diagnostic questions, this research idea occurred to me that will not require us to manually search for patterns any more. This is what is explained in the next section, which forms the basis for this research work.

CHAPTER 4

FEATURE EXTRACTION USING NATURAL LANGUAGE PROCESSING APPROACH

In the previous section, it was explained how the pattern matching rules were analyzed and developed for the questions belonging to the category of diagnostic questions. The rule based system worked well in terms of specificity and the pattern rules could be extended to other categories of the questions as well. However developing these rules manually requires a lot of time and would be very strenuous for such a large 64 number of categories.

A different approach is proposed to automatically classify the clinical questions into the different 64 categories and it is proposed that it should also work for binary classification (like diagnostic versus non-diagnostic) or multi-label classification i.e. any different number of categories. The process of finding patterns among the diagnostic versus non-diagnostic led to the observation that for some questions, we are looking for some keywords like ‘cause’, ‘differential diagnosis’. For some questions we are looking for a pair of words like ‘can’ and ‘cause’. For others we are also looking for their semantic types like symptom or syndrome or drug.

It was realized that although we are looking for all or some of these, we are not really looking for any pattern rules. We are only filling the gaps in the existing pattern of the questions that is present in the English language syntactic structure. This is explained here. Being a non-native English language speaker, and absolutely having no medical

domain knowledge, it was very arduous to analyze these clinical questions. I had to look up in the English dictionary for the part of speech of every word and then had to search online in a medical database the semantic type of the clinical words.

The English language is quite complex, and the same word can have different part of speech based on where it is placed syntactically in the sentence. For instance, the word 'that' acts as determiner in the phrase "that old woman" but as adverb in "that high", as pronoun in "What is that?" and also frequently as subordinating conjunction in the sentences. While analyzing the clinical questions, at times we would not care what the actual word is present in the question, but only about its part of speech and its position relative to other words.

In the same manner, not having any knowledge in the medical domain, I could not even differentiate between the different semantic types of the clinical words present in the questions, like symptom or syndrome. For instance, 'chest pain' is symptom, while 'heart attack' is syndrome, and I would always confuse between the two. Therefore instead of looking for the actual words, we would only look for its semantic type and fill in the gaps in the pattern based rules.

With the help of some examples, the observations in the syntactics and semantics of the clinical questions are explained below:

- "What causes pain in the coccyx area?"

In our minds, what we are actually reading in this question is 'the question word' "what", followed by a verb, in this case "causes", and then, as soon as we see the word "causes", we would not notice what the next word is, because as part of our learning process during the pattern matching analysis, we know that the next word would probably

be a noun phrase and probably some complex clinical word that we might not understand, so now we are looking up for its semantic type, in this case ‘symptom’. So now we know that it is “what causes symptom” and then we are anticipating that it would be followed by some other noun phrase which could be a clinical word like body part location, or some other noun phrase like “old man”.

Syntactically, this question is analyzed as:

{ What (pronoun), causes (verb phrase), pain (noun phrase), in the coccyx area (prepositional phrase)? }

Semantically, this question is analyzed as:

{ What, causes, symptom, body part? }

In sum, this question is like a features vector, consisting of four features, which can be represented as:

{ the question word , verb phrase with its actual words , semantic type of a noun phrase, semantic type of a prepositional phrase } : “Category”

Or,

{What, causes, symptom, body part/person} : +ve label (“diagnostic”)

The analysis of these clinical questions gave me the research idea that if we can be trained to learn the structure and pattern & categorize them, only by looking at the part of speech tags and semantic types, without being a native English speaker and having no medical domain knowledge, a computer algorithm can potentially learn this structure.

This is the basis of this research that the machine learning algorithms can be trained to label and categorize these questions, without having to a need to find any explicit pattern rules manually for the different categories of the questions, if they are

trained with proper feature vectors and that, to generate these features, only the syntactic part of speech tags and semantic types of the words are mostly needed.

As another instance, we have a question like,

"Why is she fatigued?"

This question can be represented with the same set of four features in our features vector:

{ the question word , verb phrase with its actual words , semantic type of a noun phrase, semantic type of a noun phrase/adjective phrase } : "Category"

Or,

{why, is, person, symptom} : +ve label ("diagnostic")

This would be also very contributing in that it also eliminates our need to depend on and decide whether to do a binary classification or a multi-label classification and making pattern rules based on this decision, in a supervised machine learning, because the different semantic types of the words acting as the features for the text question will enable the machine to learn the appropriate category of the clinical question. This is explained with the following example.

If a binary classification is performed (e.g. diagnostic versus non-diagnostic) for a task, these two set of questions would both be labeled as –ve in our training examples:

{"What is the dosage of drug X?", - }

{"What is the best treatment for disease X?", - }

Both of these questions when represented with our feature vector,

{ the question word , verb phrase with its actual words , semantic type of a noun phrase, semantic type of prepositional phrase } : "Category"

would be represented in the training set to the machine as:

{what, is, quantitative concept, clinical drug, - }

{what, is, therapeutic or preventive procedure, syndrome, - }

So, the machine can look at their semantic types and learn to classify them with the negative labels in the binary classification.

However, if a multi-label classification is performed, these two sets of questions would be represented in our training example set to the machine as;

{what, is, quantitative concept, clinical drug, 2112 }

{what, is, therapeutic or preventive procedure, syndrome, 2211 }

Where, 2112 and 2211 are the respective codes for their respective categories.
Now, the machine would look at their semantic types and instead would learn to classify them into their categories, and not just as positive or negative.

The same set of vectors could therefore be used effectively to do both binary and multi-label classification. We would only need to feed the machine our training set with the required labels. Thus, it eliminates our need to form different set of pattern rules for different types of classification tasks.

With this as the basis for our research theory, the biggest challenge to developing a text classification system for clinical questions would be to extract the proper features for these clinical questions using natural language processing. The more precise the features are, the better the training would be and hence the better the classification results. After the significant features are extracted in the natural language processing phase, the machine learning algorithms could then be used in a typical classifier phase to

categorize the questions into the appropriate categories or different classifier approaches could be experimented with.

In the next section, the various natural language processing techniques that are implemented for this research project are described and the current challenges that are still posed for them and what improvements are needed to be done in the future so that the feature extraction in the proposed clinical questions classification system would be more specific to the different number of question categories leading to more precise and accurate results.

SYNTACTIC ANALYSIS

The natural language processing techniques are used to extract features from the training set of clinical questions. All the natural language processing implementations were coded in Prolog and are developed with a modular approach in mind. This way a module can be configured, modified or discarded without causing any issues to other modules and works independent of other modules and would be interoperable with other programs or programming languages. The research team working on this project in the future would also then not be confined to use Prolog but could rewrite the module in the programming language of their choice.

First, the techniques were performed using a syntactic analysis of the clinical questions. It consists of the following steps.

Text Preprocessing: The text files were pre-processed to remove all double quotes (“), percentage sign (%) and other symbols or punctuation marks that are not helpful in text classification. Initially the single quote (‘) was also removed, but then it was found to be necessary for many possessive nouns and pronoun words.

Tokenization

In this step, the question text is broken down into separate words called tokens. A token can be a contiguous series of letters, digits or special character by itself. If a text word consists of letters followed by numerals, it is broken into two tokens. Whitespace characters (like any number of space, tab or return characters) separate token. At the end of this step, the question text is converted into a list of Prolog atoms (or simply, tokens).

Current Challenges: A single question text can be composed of more than one sentences or interrogative sentences. These are separated by a period (.) or a question mark. However the text frequently contains initials or abbreviations or words like 'et cetera'. So it is a current challenge to detect the end of sentence correctly. Also the text contains possessive words ('s) or embedded sentence segments followed by commas (,) or in parentheses '(', ')'. The possessive words are handled in the tagger module and the sentence segments are handled in the parser module.

Future Refinements: This module could be refined in the future to introduce a 'end-of-sentence' marker to improve and indicate the end of a sentence if a period is followed by a capital letter not followed by another capital letter. Or all the text could be downcased and then compared. Also, some foreign words from other languages occur frequently in English, so it could be refined to include foreign word characters or characters belonging to Unicode.

Part of Speech Tagging

In this step, each token is labeled with its correct part of speech using Penn Treebank. The Penn Treebank corpus is available in Prolog and has been tagged by hand by the researchers. The corpus contains the most frequent tag and a list of possible tags

for each English word. After tagging each word, few rules were constructed based on Brill transformation-rules tagging techniques (Brill 1995) for this step. With this technique, another pass is made through the list of words and for each word, two words preceding and two words following are checked for their tags to determine the context of the word, and the initial tag is then changed accordingly. The possessive words are also handled here and if it is a possessive noun (noun + ‘s) it is tagged so. If it is a pronoun followed by ‘s it is modified into two words, pronoun and the word ‘is’. If the word is not present in the Penn Treebank, it was labeled as ‘NF’ (Not Found). Separate techniques are applied for these words. The word is downcased and checked again in the corpus. The word is also checked with the WordNet database, which is also available in Prolog, though this did not have any considerable improvement. If the word is still labeled as ‘NF’ the tag is changed to noun (‘NN’) as the nouns are the most frequently occurring words in English.

Current Challenges: This step is one of the most challenging tasks for natural language processing. Many words usually have more than one part of speech, and can therefore be assigned more than one tags. The contexts of the word, its position relative to other words often determine its tag. The list was checked several times by examining its output manually to determine the correct tag and refine the rules. Also, Penn Treebank itself is not very consistent, and is said to contain errors. The Penn tags, on one hand, makes very subtle distinction in the verbs ‘VB’ (plain form) vs ‘VBP’ (the present tense form), and on the other hand, it assigns same tags to more than one part of speech, like the tag ‘DT’ for both determiners and some pronouns; or the tag ‘IN’ for both prepositions and conjunctions. Some specific words were given special rules to tag them

correctly. E.g. for the question text “Can carbon monoxide poisoning cause lethargy?” all the words after the word “can” were tagged as nouns, when the word “cause” should be tagged as verb. And it was experimented and found that none of the rules would satisfy the correct tag, so specific rule was written for such conditions.

Future Refinements: It could be experimented to use other tagging system than the Penn tags system to overcome the problem where more than one part of speech tags are assigned the same tag in Penn tags. Also other tagging corpora could be incorporated into the system. The transformation rules for tagging could be supplemented with probability based rules that assigns probability to each word and use machine learning to tag the words. This approach was not currently experimented with since pure natural language processing over machine language based processing was favored for feature extraction. The multi-word phrases could be identified and tagged with the same tag, like ‘out of breath’. Also it was found that most of the ‘Not Found’ tagged words were medical words, and therefore incorporating a medical dictionary could also improve performance. The Specialist Lexicon which is provided by the NLM, was also experimented with, however, it is still not included in the project as it has to be converted into suitable format to be used in Prolog. The accuracy of the tags depends largely on the training corpus used. E.g. The two sentences “I am keeping tabs” and “She has stimulating experiences” need the words ‘keeping’ to be tagged as verb and ‘stimulating’ as an ‘adjective’ and can be done only through the corpus, since the rules cannot differentiate the two words by looking at the surrounding words for these two words which have the similar tags in the two sentences. However the corpus could tell us that the word ‘keeping’ cannot be tagged as an adjective or vice versa.

Lemmatization

In this step, each tagged word is reduced to its basic stem form from the various morphological forms of the word. The morphological analyzer looks at the endings of the words and tries to predict the stem form and the suffix. The analyzer only looks for nouns, adjectives and verbs for suffixes as other parts of speech usually do not take suffix (e.g. the word 'perhaps' ends in –s, but it is not a suffix). After this step, the dictionary form, 'lemma' is formed from the stem with the help of lemmatizer and the lexicon from Penn Treebank. If the stem exits in the lexicon with the same tag, then it is accepted as its lemma or else it is discarded. Various spelling rules were implemented to derive the stem of the word. Also a list of irregular words was constructed that do not take any suffix (e.g. children). The output of the lemmatizer would be used for syntactic parsing as well as to generate the feature in the feature vector by passing the stem form of the words in the verb phrases. Also, for this step we are only applying inflectional morphology and not derivational morphology, i.e. the different forms of the word that have the same dictionary entry (like, 'cause' and 'causing' and not, 'causality' or 'causation').

Current Challenges: The spelling rules in English do not follow strict word formation rules and this leads to several errors in the morphological analyzer (e.g. baking, having, and panicking). The uncountable nouns do not have different morphological forms even if they end with common suffixes. (E.g. the word 'chaos' end in's', but has no suffix). Also, the same word can have more than one form, where one of the forms could be irregular form. (E.g. the word 'eat' has also an irregular form 'ate').

Future Refinements: This module can have many more refinements in the future by looking at the spelling and word formation rules carefully. A more comprehensive list

of irregular forms can be constructed. Also, in addition to suffix, prefix rules can be applied. If required, derivation morphology can also be done to help with the semantic analysis of the text.

Syntactic parsing

In this step, each tagged word is labeled with a linguistic label stating its part of speech and then the labeled words are grouped together into phrases or constituents which are also labeled. Each of these phrases thus can be built into a tree that describes the structure of the sentence. The parsing is done so that the computer can recognize a sentence structure. Each of the phrases in the sentence structure could be the answer to a question. Each phrase can be replaced by a word or another phrase, or moved at a different position within the structure. The parsing technique consists of a set of rules, called grammar, and an algorithm to process these rules. This set of rules say how the tree can be built, and are written to describe the language itself. The syntactic parsing in natural language processing is a huge research topic in itself, and a complete treatise of this subject is beyond the scope of this document.

For this project, a top-down parser is implemented. The top- down parser starts with the root of the tree and then parses down the tree from the root to the leaves, which are words themselves. The subject of this research project is concerned with sentences that are mainly questions. Therefore, many specific grammar rules were written to parse the questions. However, in the training set, many of these question texts have both a sentence and a question in the same instance, and therefore the parser is needed to work for both types of sentences.

It was decided to implement chunk parsing instead, using the top down parser, rather than the complete sentence structure. Chunk (shallow) parsing involves finding the basic structures among the sentence like noun phrases, verb phrases, prepositional phrases in the sentence, rather than building the complete tree structure. This technique originally comes from the idea that human brain processes long complex sentence in chunks of structures and we even pronounce a sentence in chunks where the intonation varies with these chunks.

The chunk parsing is explained below with the help of a few different example questions parsed by the code in this project.

Input: "What is the cause of this long term memory loss?"

Output:

```
[chunk,[qs(['What'/'WP'])]],[chunk,[vp([v([(is)/'VBZ'])])]],[chunk,[np(
[d([the/'DT']),n([cause/'NN'])])]],[chunk,[pp([p([of/'IN']),np([d([this
/'DT']),adjp([adj([long/'JJ'])]),n([term/'NN',memory/'NN',loss/'NN'])])
])]],[chunk,[separator([ (?)/'.'])]]
```

Analysis: This text question is broken down into four chunks.

`[chunk,[qs(['What'/'WP'])]]` is a chunk that describes the start of a question sentence by marking it with 'qs' followed by the question word 'What' and its Penn tag 'WP' which is an interrogative pronoun.

`[chunk,[vp([v([(is)/'VBZ'])])]]` is a chunk that describes a verb phrase by marking it with 'vb' which consist of a linguistic structure verb, marked by 'v' followed by the word 'is' and its Penn tag 'VBZ' which is a 3rd person singular present verb.

`[chunk,[np([d([the/'DT']),n([cause/'NN'])])]]` is a chunk that describes a noun phrase by marking it with 'np' which consist of a linguistic structure composed of

structures – a determiner and a noun, marked by ‘d’ and ‘n’ respectively, followed by the words ‘the’ and its Penn tag ‘DT’ which is a determiner, and the word ‘cause’ and its Penn tag ‘NN’ which is a noun.

`[chunk,[pp([p([of/'IN']),np([d([this/'DT']),adjp([adj([long/'JJ'])]),n([term/'NN',memory/'NN',loss/'NN'])])])]]` is a long chunk that describes a prepositional phrase by marking it with ‘pp’ which consist of a linguistic structure composed of structures – a preposition and a noun phrase, which itself is composed of a determiner, an adjective phrase (which itself is composed of an adjective structure) , and a noun structure, all these marked by ‘p’, ‘np’, ‘d’, ‘adjp’, ‘adj’ and ‘n’ respectively, followed by the words ‘of’ and its Penn tag ‘IN’ which is a preposition, the word ‘this’ and its Penn tag ‘DT’ which is a determiner, the word ‘long’ and its Penn tag ‘JJ’ which is an adjective, the words ‘term’, ‘memory’, ‘loss’ and their Penn tag ‘NN’ which is a noun.

`[chunk,[separator([ (?)/'.'])]]` just marks the end of a question or a sentence.

The code currently can parse both a sentence and question and indicate which one is a question. As an example question text,

Input: “Chronic abdominal pain, don't know what to do for it. What is the cause?”

Output:

```
[chunk,[adjp([adj(['Chronic'/'JJ'])])]],[chunk,[np([adjp([adj([abdomina
l/'JJ'])]),n([pain/'NN'])])]],[chunk,[conjunction([ (',')/
(',')])]],[chunk,[vp([v(['don\'t'/'VB',know/'VB'])])]],[chunk,[np([pron
oun([what/'WP'])])]],[chunk,[pp([p([to/'TO'])])]],[chunk,[vp([v([do/'VB
P'])])]],[chunk,[pp([p([for/'IN']),np([pronoun([it/'PRP'])])])]],[chunk
,[separator(['.'/'.'])]],[chunk,[qs(['What'/'WP'])]],[chunk,[vp([v([
(is)/'VBZ'])])]],[chunk,[np([d([the/'DT']),n([cause/'NN'])])]],[chunk,[
separator([ (?)/'.'])]]
```

The input text has two sentences. The first sentence of the input text is a sentence and the second is a question. Also there are two ‘what’ words in the input text. As seen from the output, the first chunk is not marked by ‘qs’, nor does the chunk containing the first ‘what’ word, but only the chunk containing the second ‘what’ word is marked by ‘qs’ indicating the start of a question. Also the embedded sentence “don’t know what to do for it” is indicated by the structure ‘conjunction’. Also the apostrophe in “don’t” is handled by the code.

Current Challenges: There are currently many challenges associated with the parsing technique as it is a very complex process and is a huge research topic. Often the sentence includes structural ambiguity, which cannot be solved by the parser. Someone other than parser has to decide it. E.g “Oxygen saturation fine” actually misses the word “is” and it’s better to have a different style of writing or speaking. Also certain verbs can only take certain types of objects and not others, so it is a challenge on how to group words together.

Future Refinements: “I don’t know what to do for it” is currently marked as a sentence, but in effect is actually a question. This comes under the topic of pragmatics where the context of the spoken text decides its parsing structure. Additional rules could be written for sentences like “Oxygen saturation fine”. Another solution is the parser gives alternatives, and then additional semantic interpretation is applied on them. Also the technique of subcategorization can be used to decide what words or phrases take what kind of objects with them. Another technique that could be applied is to make another pass at parsing after the first pass, to correct the incorrect parsing. It could be

experimented to use probabilistic lexical parser that learns from its previously parsed examples.

SEMANTIC ANALYSIS

As stated earlier, there are many questions belonging to different categories that are syntactically similar but semantically different. It is necessary to incorporate some semantic features into the data set. For instance,

"Can symptom x cause disease y?"

"Can disease y cause symptom x?"

"Can drug x cause finding y?"

The three questions are all syntactically similar and have the same parsed output. Without the semantic analysis, it is impossible to define which a diagnostic question is or which one is a treatment question.

For semantic analysis of a clinical question, we would need a knowledge base or ontology in English that can understand these structures and give us the semantic interpretation to allow us to differentiate between the different categories of the questions. Initially it was experimented to use WordNet, but it does not have many medical terms to allow us a good enough semantic analysis to differentiate between these questions. Therefore, we need a domain specific knowledge base that can help us in the semantic analysis of the restricted domain-specific complex words. A comprehensive medical database, dictionary, knowledge base or resource can help us categorize the words into its appropriate semantic types. This is explained in the next chapter.

It is also to be noted that the semantic analysis would require us to find only those words from sentences that add meaning to our feature vector. E.g. "there is a rash", the

word “is” does not add anything meaningful to our feature vector. Therefore the phrases that are not semantically relevant should be excluded from our features vector. After observing several questions examples manually, it was decided that copula (like is, are, were), adverbs (more, fast, quickly), pronouns and possessive words can be safely excluded from the feature vector. More research in the future is needed to carry out the semantic interpretation of these words and phrases.

CHAPTER 5

SEMANTIC ANALYSIS USING A MEDICAL KNOWLEDGE BASE

THE UNIFIED MEDICAL LANGUAGE SYSTEM

The question classification system would need some dictionary database to correctly identify the terms in the question as belonging to either the 'drug' category or 'disease' category, for instance. Among the few biomedical databases available, the Unified Medical Language System (UMLS) is the largest comprehensive database of biomedical terms and concepts, their synonyms and the relationships between them, and is provided by the National Library of Medicine (NLM) for the interested researchers to access for their use. The UMLS (Lindberg 2001) consists of three parts – the Metathesaurus, the UMLS Semantic Network and the Specialist Lexicon & Lexical Tools. There are also various tools available with the UMLS which are open-sourced, to browse through the database or for further use in the application development.

The Metathesaurus is a collection of various biomedical vocabularies and thesauri and contains the concepts, their meanings and also the original sources to which they belong. One concept could therefore actually be mapped to different categorizes based on the various sources from which the concept is mapped. The UMLS Semantic Network contains the set of various semantic types and the relationships that may occur between these types. Thus, for all the concepts defined in the Metathesaurus, the Semantic Network holds the hierarchical categorization for these concepts and the relationships between them. There are 133 semantic types and 54 semantic relationships defined in the

Semantic Network. However, these semantic types are also broadly grouped together into 15 different semantic groups. The UMLS Specialist Lexicon and Lexical Tools is a lexicon of English words and biomedical terms with their various morphological forms and natural language processing tools to process them.

Bodenreider (2001, 2004) shows the various complex relationships that exist in the Metathesaurus, and how one can detect and prevent circular hierarchical relationships. Dai et. al. (2008) however showed the problems with the UMLS mapping program and developed their own version which proved to work much faster using Open Biomedical Ontologies.

METAMAP

For this project, MetaMap program (Aronson 2001, 2010) is used. MetaMap is a program provided by the National Library of Medicine (NLM) and is a widely used program to extract semantic features from UMLS and can be used to link the concepts in the UMLS. MetaMap is an open-sourced program written in SICStus Prolog. It can be given complete text or phrases of text and the program will output all the possible combinations of the text phrases with their semantic types. It can generate human readable text output or a Machine Output (MMO) of Prolog listings of the terms.

A sample Machine Output (Lang 2014) when given the phrase ‘Denied Chest Pain’ as input, is a predicate ‘mappings’ which contains lists of the structures ‘map’ of all the possible combinations of the text phrases with their semantic types as:

```
mappings([
map( -901 ,
[ev(-660,'C0332319','Denied','Denied (qualifier)',[denied],[qlco],
[[[1,1],[1,1],0]],no,no,['MTH','SNMI','SNOMEDCT','CHV'],[12/6],0),
 ev(-901,'C0008031','Chest Pain','Chest Pain',[chest,pain],[sosy],
[[[2,3],[1,2],0]],yes,no,['ICF','ICD10CM','CCS'],[19/10],0)] ),
```

```
map( -901 ,
[ev(-660,'C0332319','Denied','Denied (qualifier)',[denied],[qlco],
[[[1,1],[1,1],0]],no,no,['MTH','SNMI','SNOMEDCT','CHV'],[12/6],0),
 ev(-901,'C2926613','Chest pain','Chest pain',[chest,pain],[clna],
      [[[2,3],[1,2],0]],yes,no,['LNC','MTH'],[19/10],0)] )
```

The machine output labels 'Denied' as of semantic type Qualitative Concept (qlco) and 'Chest Pain' as the semantic types Sign or Symptom (sosy) and Clinical Attribute (clna). Thus, 'chest pain' has been categorized into two semantic types. In another instance, when given the input "Heart Attack", the Machine Output is:

```
ev(-1000,'C0027051','Heart attack','Myocardial
Infarction',[heart,attack],[dsyn],[[[1,2],[1,2],0]],yes,no,['AOD','CHV'
,'CSP','CST'], [0/12],0,0),
ev(-861,'C0018787','Heart','Heart',[heart],
      [bpoc],[[[1,1],[1,1],0]],yes,no,['AOD','CHV','CSP','FMA'],[0/5],0
      ,0),
ev(-861,'C0277793','Attack, NOS','Onset of illness',[attack],
      [tmco],[[[2,2],[1,1],0]],yes,no,['AOD','CHV','CSP','MTH'],[6/6],0
      ,0),
ev(-861,'C1261512',attack,'Attack behavior',[attack],
      [socb],[[[2,2],[1,1],0]],yes,no,['AOD','CHV','MTH'],[6/6],0,0),
ev(-861,'C1281570','Heart','Entire heart',[heart],
      [bpoc],[[[1,1],[1,1],0]],yes,no,['MTH','SNOMEDCT_US'],[0/5],0,0),
ev(-861,'C1304680','Attack','Observation of attack',[attack],
      [fndg],[[[2,2],[1,1],0]],yes,no,['CHV','MTH','SNOMEDCT_US'],[6/6]
      ,0,0)
])
```

Here, the machine output labels 'Heart Attack' as of semantic types Disease or Syndrome (dsyn), Body Part, Organ, or Organ Component (bpoc), Temporal Concept (tmco), Social Behavior (socb), and Finding (fndg). Thus, 'heart attack' has been categorized into five semantic types. But in order to correctly use this as a feature for our question classification categories, the only semantic types relevant out of the five outputted are Disease or Syndrome (dsyn), and Finding (fndg).

Thus, it becomes a task to identify and select the relevant semantic types out of the 133 semantic types present in the UMLS for each question category to use as features and discard the other semantic types. But even then, since 'heart attack' could be related

to both semantic types ‘Disease or Syndrome’ and ‘Finding’, it becomes important to use this feature in combination with the other features extracted using syntactic and semantic analysis on the other textual words in the question.

It is to be noted that MetaMap is a huge program requiring around 8 GB of hard disk space, 2 GB of RAM and has around 40 MB of source code text files.

INCONSISTENCIES

The MetaMap source code is written in Prolog, and so it was considered to study its source code and integrate it directly into the project’s source code. Therefore, a considerable amount of time was spent to study the source code. Unfortunately the source code is neither well documented nor is commented very well that would help in understanding it. It was found however that the source code uses some C libraries in the background for the mapping engine and much of the data in the knowledge source was stored in relational tables (Aronson 1996). Also it uses Sicstus Prolog, which is only commercially available. It was found that MetaMap also provided Java API to embed with your software code. Experiments were also conducted with the Java API, however due to the large size of the program, the running times were very long. Finally, it was decided to call MetaMap with the command line option and integrate it with the code.

MetaMap provides a number of data options, processing options and output options to use as per the requirement of the project. For this project, the output option was chosen that gives output in the form of Prolog predicates which can then be directly called from your source code. Also, a number of experiments were conducted with these options to find the correct set of options to be used for the project. Not only do these options change the processing of the data, and hence also its output and the semantic type

required, but also the running times and memory usage. With some options it would run for hours before any output is shown and several times the computer system would hang up showing blue screen error due to huge memory usage. The experiments then have to be run again all from the start.

During this process, the MetaMap output was checked to determine the relevant semantic types and how to tune it best to get the best overall performance. Since our proposed system requires that the answer to the clinical questions should be available in minutes, the proper tuning of the program is critical to our system.

There are many inconsistencies in the MetaMap that leads to a wrong semantic type resulting in the wrong classification of the question. In this step, each word was tagged with its semantic type and checked for inconsistencies in it. This can be improved by manually reviewing it. This is explained below with an example. When MetaMap is given the input question text as:

"Oxygen saturation is fine. Why is she short of breath?"

The output (truncated) is as shown below:

```
metamap14.binary.x86-win32-nt-4 (2014)

       phrase('she short of breath?')
       map(-733,[
       ev(-770,'C1822717','SHE','SHE gene',[she],[gngm],[[[1,1],[1,
       1],0]],yes,no,['HGNC','MTH','OMIM'],[34/3],0,0),
       ev(-
       770,'C1282927',short,'Shortened',[short],[qlco],[[[2,2],[1,1],0]]
       ,yes,no,['CHV','MTH'],[38/5],0,0),
       ev(-
       770,'C0225386','Breath','Breath',[breath],[bdsu],[[[4,4],[1,1],0]
       ],yes,no,['LNC','SNO
       MEDCT_US'],[47/6],0,0)]),
```

Thus, instead of getting the semantic type as 'sosy', sign or symptom, it outputs the type as 'bdsu', body substance.

When instead it is given the phrase “shortness of breath”, the output is as shown below:

```
[ mappings([
            map(-1000,[ev(-1000,'C0013404','Shortness of
Breath','Dyspnea',[shortness,of,breath],[sosy],[[[1,1],[1,1],0],[
[2,2],[2,2],0],[[3,3],[3,3],0]],yes,no,[
'CHV','CSP','CST','DXP','ICD10CM','ICPC','MSH','MTH','NCI','NLMSu
bSyn','OMIM','S
NOMEDCT_US'],[0/19],0,0)])
```

As seen above, the phrase is now correctly classified as ‘sosy’, sign or symptom.

The various options were experimented with and it was found that the best settings for the MetaMap for our project would be

```
Control options:
  composite_phrases=4
  lexicon=db
  mm_data_year=2014AA
  machine_output
  allow_concept_gaps
  term_processing
  allow_overmatches
  no_derivational_variants
  prune=100
  threshold=700

mappings([map(-944,[ev(-847,'C1830531','When short of
breath','When short of
breath',[when,short,of,breath],[fndg],[[[1,2],[2,3],0],[[3,3],[4,
4],0]],yes,yes,['L
NC'],[0/15],0,0)])]).
```

With these settings, the output is shown as ‘fndg’, finding and the phrase ‘short of breath’ is mapped to the phrase ‘When short of breath’. These settings tell the MetaMap to take the input as a single term, but allow the use of overmatches, or concept gaps, and then prune the results to the top 100 and only evaluate those candidates that have a score of above 700. MetaMap also allows you to restrict the database sources, such as MESH, or SNOMED-CT, however after experimenting with several text questions, it was found

that the text was not mapped to any semantic type when restricted to these sources, and therefore it is not used in later implementations.

MAPPING TO THE SEMANTIC TYPES

In this step, each tagged word is mapped to its semantic type by the use of MetaMap program. The output of the MetaMap used is Prolog Machine Output, and it is quite detailed. Therefore at this stage, only the semantic types from the output were used. The code can directly call the Prolog predicates in the MetaMap output so as to extract the semantic features for the questions and integrate them with the features for the feature vector into a machine learning package.

It becomes a manual task to identify and select the relevant semantic types out of the various semantic types, which are output by MetaMap, to use as features and discard the other semantic types. Therefore it was decided to do a flat level classification for the initial prototype stage, and allow the machine to learn the relevant semantic types from the feature vector for each of the 64 categories of the question. As explained in the previous section, the machine could be allowed to learn to categorize into a binary, or multi-label classification depending on how the training data set is labeled. After the concepts are mapped to its semantic types and are written to an external file, they are integrated into the feature vectors for the machine classification task, which uses them to train on its training set of clinical questions. This is explained in detail in the next chapter.

CHAPTER 6

QUESTION CLASSIFICATION WITH MACHINE LEARNING

The machine learning techniques were applied with the notion of dividing the techniques into two groupings – Generative model and Discriminative model (Srihari 2010). The generative model builds a joint probability distribution on all the variables – both input and output, like Naïve Bayes, Bayesian network, Hidden Markov Model. The discriminative model estimates posterior probabilities directly based on the observations, or input variables, like k Nearest Neighbour, Artificial Neural Network, Decision trees.

As discussed previously, there are broadly two approaches to follow for classifying the clinical questions into the given 64 categories. The first approach is to do a flat classification on the questions into one of the 64 categories. The second approach is to do the classification at the hierarchical level – classifying the questions into the 5 top-level categories and then after the top level (primary) classification, classifying the questions further into deeper levels within each of these top levels, following this approach for all the four levels of hierarchy.

However, at this stage of the project, the hierarchical classification was not researched as it was considered, because it would be required to manually classify the semantic types out of the total 133 types for each of these levels. Also, it was proposed that the system could be allowed to learn to categorize into the appropriate categories based on the training sample labeling, the project was implemented with only the flat classification into all the 64 categories of the questions. It is hoped that once the initial

proposed work gains some more progress, the research teams working on this project in the future could experiment with the hierarchical classification in depth.

It was however considered to apply the machine learning techniques with two different approaches in mind. One is to do a flat classification into the 64 categories without using any semantic features and only using the various statistical text measures like tfidf, unigrams, bigrams. The other approach is to do the same flat classification with the semantic features extracted from the previous majority part of the research project.

APPROACH 1 – FLAT CLASSIFICATION WITHOUT USING SEMANTIC FEATURES

In this approach, the questions are classified at a flat level into one of the 64 categories of questions only using the various statistical textual measures. Following machine learning algorithms were tried –

1) Naïve Bayes Classifier

Naïve Bayes is a very commonly used machine learning method for text categorization. It is based on Bayes' Law. It uses the assumption that the probability of observing one word in a given text document is independent of other words in the document and thus calculates the probability for each label using the words as features. For our dataset of questions, this assumption is not entirely true. For example, in a diagnostic question, it is more likely that 'what' and 'cause' come together.

2) Artificial Neural Network

Artificial Neural Network is a method that simulates the neural network processing in living beings. It consists of input nodes connected to the output nodes via hidden nodes in network with weights attached to each edge connecting the nodes. A

very commonly used algorithm is Backpropagation using sigmoid function as the activation function to update the weights based on the previous inputs. Artificial Neural Network can be used to represent complex relationships and be used for discrete and continuous valued data.

3) k-Nearest Neighbor

k-Nearest Neighbor is a lazy learner as it delays the determination of a trained model until it sees a new instance, which is then classified by taking the label of the majority (or average) of its 'k' number of nearest neighbors. The testing time is therefore long as for each new instance it has to recalculate the 'k' nearest neighbors.

4) Decision Tree

Decision tree is an algorithm that classifies the instance using 'if-then' rules starting from its root node represented by the various attributes values or features, following a hierarchical path to its leaf node, which represents the final label. Decision trees can only be used for discrete valued attributes, however can handle noise better. Pruning can be used for error handling and overfitting.

5) Maximum Entropy Classifier

Maximum Entropy Classifier uses logistic regression model and has gained popularity for text classification. It is a counterpart of Naïve Bayes without the assumption that occurrence of one word (feature) is independent of other words (features) for a given label. It works by iterating through the model multiple times to update the probabilities (weights) using maximum likelihood and is therefore slower than Naïve Bayes.

6) Support Vector Machine

Support Vector Classifier is based on an algorithm that finds a linear model, using the maximum margin hyperplane. It does this my mapping the non-linearly separable examples into a hyperplane, and among all the hyperplanes possible, the maximum margin hyperplane is the one that is the perpendicular bisector to the shortest line connecting the planes containing the positive and negative examples to classify them. Support Vector classifier has gained huge popularity for the text classification tasks.

EXPERIMENTS AND RESULTS WITH APPROACH 1

Initially experiments were performed with the open source WEKA package for the various classification algorithms.

Text Preprocessing: The machine learning experiments were performed on the general questions from the training set and the testing set. The text files were pre-processed to remove all single quotes(‘), double quotes(“), and percentage sign (%) as Weka showed error inputting the file with these characters in the input file. The text was then processed by running the ‘NominalToString’ filter. Then, the text was processed to convert into numeric attributes to tf*idf values by using ‘StringToWordVector’ filter with ‘LovinsStemmer’ and ‘IDF-Transform’. The resulting file has 2562 instances with 3903 attributes and 64 target class labels.

Weka implementation of k-Nearest Neighbor, rules.NNge (kNN with Non-nested generalization) was implemented on the training set. It took a very long time to get the results and generated long sequences of rules. When the classifier was run on 10-fold cross validation, it seemed to run forever and finally the Weka had to be stopped after long hours.

The similar problem was faced with the Weka implementation of ANN; Multilayer Perceptron with the default settings was run on the data set, but the algorithm seemed to run forever, but after 8 hours when there was no result, it had to be manually stopped. It was run again with some different settings, but encountered the same problem.

The biggest challenge to running these algorithms was not Weka, but rather the large number of attributes. With around 4000 attributes for multi-label classification (64 categories), complex rule generating algorithms cannot work well. Also several tools were researched that could be especially used for text classification. A different package was also used called MALLET (McCallum 2002) which is also open source and especially built for text classification. However, MALLET lacks the diverse number of algorithms and experimentation approaches that are present with Weka.

The results can be summarized in the Table 3 below:

Table 3: Results with Flat Level Classification without Semantic Features

	Naïve Bayes	**Max Entropy**	**C45**	**Support Vector**
Training				
Accuracy Mean	0.5691	0.9719	0.7056	0.4679
Standard Deviation	0.0054	0.0020	0.0055	0.1227
Standard Error	0.0017	6.44E-04	0.0017	0.0304
Testing				
Accuracy Mean	0.3687	**0.4667**	0.3418	0.3762
Standard Deviation	**0.0171**	0.0238	0.0306	0.1311
Standard Error	**0.0054**	0.0075	0.0096	0.0347

As seen from the tables above, the highest result in the testing set was obtained with Maximum Entropy classifier with accuracy of 46.67%. However Naïve Bayes generated the lowest standard error and the standard deviation while the Support Vector showed the highest error which was surprising. There are 64 categories of questions,

considering equal distribution of data, the baseline probability is 1.56%. However, the questions in the data set are not uniformly distributed, and it was observed that the class type 2.1.2.1., i.e. the generic type question "What is the drug of choice for condition x?" has the highest number of instances equal to 323 out of the total 2562 distinct instances. Given this fact, the baseline accuracy is 12.61%.

It was found that running the machine learning algorithms to do a flat classification using textual statistical measures, into one of the 64 types, performed above the baseline but the accuracy obtained by these algorithms was still low.

Since there are many questions belonging to different categories that are syntactically similar but only semantically different, it is necessary to incorporate semantic features into the classification task. Thus, feature extraction becomes the key factor and the most challenging part for this system so as to extract the semantic features to be used with the syntactic features in the classification of the clinical questions.

APPROACH 2 – FLAT CLASSIFICATION USING SEMANTIC FEATURES

As described in the previous sections, the natural language processing techniques on both the syntactic level (part-of-speech tagging, parsing) and the semantic level (extraction of noun phrases and nouns, identifying them as disease/syndrome using ontology) are used to extract the relevant features from the clinical questions. Then using them as the features values for different categories of the questions, machine learning techniques would be used to appropriate classify them into the correct types.

Feature Generation With Semantic Analysis

As stated earlier, there are many questions belonging to different categories that are syntactically similar but semantically different, it is necessary to incorporate some

semantic features into the data set, for instance, if we take “what cause” questions as positive label, and the rest all as negative label, the following three types of questions should be classified accordingly as:

{“Can symptom x cause disease y?” , + }

{“Can disease y cause symptom x?” , - }

{“Can drug x cause finding y?” , - }

Since the above training instances are all syntactically similar, the semantic features would only help into the correct labeling of these instances. So, the feature vectors are generated for all 64 types of the clinical questions using the semantic types of the concepts found in the questions. So “chest pain” could be made into a feature ‘symptom’ while “cancer” as ‘disease’.

For this step, the feature vector for each question text was made of four features as described in the previous chapter. The features are the question word (like, what, why, how, is, and so on), the words present in the verb phrases (like, cause, is, would, etc.), the semantic type of the noun phrase, and the semantic type of the second phrase that could be a prepositional phrase, or adjective phrase. After analyzing several questions, it was considered to choose the last phrase occurring in the sentence as the second phrase whose semantic type should be used. Thus, our feature looks like this

{question word, string of words in the verb phrase, semantic type of the noun phrase, semantic type of the last phrase}

Therefore, with this feature set, only the words from the verb phrase would be generated into the statistical textual features by machine learning techniques, but the rest three features are nominal features that take values from a set of values. For the question

word feature, the set of values is {Are, Can, Could, Do, Does, Has, Have, How, Is, Should, What, What's, When, Where, Who, Why, Will, Would}. The semantic type features take values from the set of 133 semantic types defined in the UMLS.

EXPERIMENTS AND RESULTS WITH APPROACH 2

For this step, the performance is evaluated on the general questions from the training set and the testing set, into the various 64 categories. The words from the verb phrases were processed to convert them into numeric attributes to tf*idf values by using 'StringToWordVector' filter with 'LovinsStemmer' and 'IDF-Transform'. The resulting file has 2580 instances with 327 attributes and 64 target class labels. The testing file had 1077 instances with 242 attributes. The training was done with 10- fold cross validation.

Since we did not have success with the Weka implementation of k-Nearest Neighbor, rules.NNge, in the approach 1, for this approach instead, another implementation for the same, lazy KStar was experimented. Also there was an error encountered with the Maximum Entropy algorithm implementation using Weka's multinomial logistic regression, and is further looked into, as to its cause for hang up.

The results are summarized in the Table 4 below:

Table 4: Results with Flat Level Classification Using Semantic Features

	Lazy K-Star	**Naïve Bayes**	**J48(C45)**	**Support Vector**
Training				
Accuracy Mean	0.5205	0.3895	0.4127	0.4914
Standard Deviation	0.1001	0.111	0.1113	0.1226
Standard Error	0.0175	0.0214	0.0209	0.0303
Testing				
Accuracy Mean	0.0380	0.4131	0.4661	**0.5506**
Standard Deviation	0.0124	0.1093	0.1093	0.1226
Standard Error	0.0308	0.0209	0.02	0.0303

The algorithms performed with better results than in the approach 1, which had no semantic features. The highest performance was shown by the Support Vector classifier with 55% accuracy. However, Lazy K-star really underperformed, with only 3% accuracy with the testing set. Naïve Bayes and J48 had the same standard deviation error.

Other Experiments: A few other initial experiments were also performed on the above training and testing set, using the semantic features. In the first set of these other experiments, it was tried to perform hierarchical classification at the two levels. On the first hierarchical level, the questions were first classified at the top 5 levels. After this step, on the second hierarchical level, the questions were further classified into the 64 types within each of the top five levels. Thus, there were five more models at the second hierarchical level to classify these questions deeper into the further categories below within each of these top five levels. However, the results obtained were not as good as the flat level classification. This suggests that more strategies at the hierarchical classification are needed.

In another set of experiments, it was attempted to again perform the hierarchical classification at the two levels but this time, at the top hierarchical level, it was tried to classify the questions into the top three most frequent categories of the questions. Thus, at the first hierarchical level, we have four target class labels – three labels for the top three most frequent categories and the fourth label for collectively all other categories. At the second hierarchical level, the fourth label categories from the above level were classified. The highest accuracy reached was 76.88% with Support Vector classifier at the first hierarchical level. But the results for the second level obtained were again not as good as flat level classification. The results are further discussed in the next section.

CHAPTER 7

DISCUSSION

The results with the rule based pattern matching module show that it performed very well on diagnostic questions with respect to specificity, reaching around 96% but the recall was very low for short questions and not high even for the general questions. This is a tradeoff where the rules that are incorporated to increase the specificity, lower the recall. The precision for the short questions reached up to 70%, but for a binary classification, it is quite possible to have better results even with less complicated rules.

Approach 1 with the machine learning to perform a flat classification on 64 categories of questions using statistical textual measures show that classifying at the flat level leads to generally a low accuracy. It was found that the machine learning algorithms performed above the baseline but the accuracy obtained by these algorithms was still low, the highest result was obtained with Maximum Entropy classifier with an accuracy of 46.67% for all the 64 categories of the questions. The number of classification labels is quite large in this case and more sophisticated approaches are needed. Also since the number of features that were generated using the statistical measures was enormous, the algorithms that produce complex rules tends to perform very slowly and poorly.

It is hoped that even with such large number of features using statistical measures, the results could be improved if the classification is performed on the hierarchical level. After the classification on the top level, the classification could be done within each of these levels. But it would then pose the challenge of having to classify on a very

distinctive features since within the level, the questions could be very similar syntactically but differs only in semantics. It was therefore proposed to incorporate semantic features into the classification task.

The approach 2 with machine learning using semantic features show that the results showed an improvement for about almost 10% increase in accuracy, which although not impressive, are not bad either. The Support Vector Classifier had the highest accuracy at 55% for the 64 categories, which is comparable to the rule based pattern matching binary classification result. As stated earlier, with such a large number of categories, the baseline accuracy is itself only 1.56%. Taking this into consideration, they still performed much better than in approach 1. Also given the fact that the approach - 2 has words from verbs that were transformed to tf-idf values, resulted in 327 attributes.

It is to be also noted that the algorithms performed better on the testing set than the training set proving that after the initial learning, the machine is able to find patterns in the unknown data set. The lack of questions corpus of suitably large size also lowers the accuracy during the learning stage.

The number of attributes also considerably got reduced. The approach 1 has 3903 attributes while approach 2 only has 327 attributes, which is less than 10% of the original number of attributes. It could also be experimented to not convert the words from verb phrases into statistical measures, but used only as binary features. Because many of the categories have the same words, the TF-IDF transform makes the impact of this presence less significant.

The features were carefully analyzed to better understand the results. And it was found that they had many of the semantic types that were mapped to very general

categories, like patient, diagnosis, etc instead of having specific unique values. Also the phrase selection and the extraction of the noun phrase could have considerable impact on the performance. Since during this experiment, the number of phrases to use was limited to two, the length of the feature vector was kept limited at four, it could be useful to be flexible with this limitation.

. The integration of MetaMap with the program should have considerably improved the performance. However there are many inconsistencies in the MetaMap as explained in the previous sections that leads to a wrong semantic type resulting in the wrong classification of the question. This can be improved by manually reviewing it. Also MetaMap evaluates the many possible candidates with an evaluation score but the final mapping score is different from the evaluation score, which could also be taken into consideration while selecting the semantic type

In general, it is always a difficult criterion to choose between the flat and hierarchical classification. The initial experiments with the two-level hierarchical classification with five labels at the top level did not generate good results. For the topmost level of hierarchical classification, it could also be considered to classify the questions into diagnostic questions vs. the rest other questions, instead of classifying into five labels. It would be normal to expect that the binary classification would yield better results than multi-label classification. But there is a higher error propagation chance with this approach, and so it could be also considered to do both flat classification and hierarchical classification at the same time and then comparing both the results to give the final majority labeling.

Also, as it was found that approximately 80% of the clinical queries can be found in only 20% of the question types, another approach considered is to focus on classifying the questions correctly into the top three most frequent question types appearing in the data set. The results obtained with this classification strategy yielded around 77% accuracy at the top level, but not as good results into the second level. So it could be experimented to classify into the top 10 categories, instead of the top 3, which accounts for 80% of the clinical questions. The three most frequently occurring question types are:

class type 2.1.2.1. “What is the drug of choice for condition x?” (11%),

class type 1.1.1.1. “What is the cause of symptom x?” (8%), and,

class type 1.3.1.1. “What test is indicated in the situation x?” (8%).

Besides the top three question types, the other most frequently occurring question types are –

class type 1.1.2.1. “What is the cause of physical finding x?”,

class type 1.1.3.1. “What is the cause of test finding x?”,

class type 2.1.1.2. “What is the dose of drug x in situation y?”,

class type 1.2.1.1 “Can condition y cause finding x?”,

class type 2.2.1.1 “How should I treat condition y?”, and,

class type 3.1.1.1. “How should I manage situation y?”.

The classification task of the questions could also be focused mainly with the experimentation with different classifier approaches, algorithms and methods, which is a different machine learning research topic in itself and is beyond the scope of this thesis, and would be a very good research topic for future research teams working on this clinical questions classification project.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

The results show that classifying at the flat level leads to generally a low accuracy and it is hoped that the hierarchical level text classification could increase the results. As can be seen from the results with a second approach, preliminary tests were done using semantic analysis by integrating MetaMap with the project and using the syntactic and semantics features to use as feature values with the various machine learning algorithms and it was found that the program was able to classify better than the previous results.

This project work should not be taken as an end product but the aim of this thesis was to propose and determine the feasibility of a clinical questions classification system that gives precise and accurate answers by classifying the questions correctly into 64 specific categories of questions, which is a daunting task for humans itself. The results improved after using the semantic features and achieving more than 50% accuracy proves the feasibility of the system, and this gives an initial prototype for the research teams to work on this project in the future.

The future directions of the thesis project are to improve the semantic analysis that results in high recall, specificity and precision values. It is certainly hoped that the question classification system with semantic analysis would yield better results by appropriately classifying the clinical questions into correct categories. Further features can be developed for each of the 64 separate categories so that it would be useful to classify within each of the levels after the top level hierarchical classification. Also, it can

be researched if some of the categories from these 64 categories could be merged to reduce such a large number of categories and thus reduce the challenges of this questions classification task.

It would also be useful to refine the taggers and parsers especially adverbs, particles and gerunds. It is suggested that using an active chart parser or shift reduce parser with Oracle can significantly improve performance. Also, if the word is not found in corpora, the user could be asked to tag the word with appropriate part of speech. Further the program can be extended by using other online Medical dictionaries like Specialist Lexicon and incorporating techniques like Named Entity Recognition, Word Sense Disambiguation and Coreference.

REFERENCES

Agrawal S., Yu H. (2010). Biomedical Negation Scope Detection with Conditional Random Fields. *J. Am Med Inform Assoc*. 17(6): 696-701.

Aronson, A. (1996, February 2). MetaMap Technical Notes. Retrieved from https://ii.nlm.nih.gov/Publications/Papers/metamap.tech.pdf.

Aronson, A. (2001). Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. *Proceedings of the American Medical Information Association Symposium* 2001:17-21.

Aronson, A. R., & Lang, F. M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association* 17(3): 229-236.

Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics* 22(1):39-71.

Bodenreider, O. (2001). Circular hierarchical relationships in the UMLS: etiology, diagnosis, treatment, complications and prevention. *Proceedings of the American Medical Information Association Symposium* 23:57-61.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research 32*(suppl 1):D267-D270.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, *21*(4), 543-565.

ClinicalQuestions Collection (2013, January). Retrieved from https://clinicaltrials.gov/

Dai, M., Shah, N. H., Xuan, W., Musen, M. A., Watson, S. J., Athey, B. D., & Meng, F. (2008). An efficient solution for mapping free text to ontology terms. *AMIA Summit on Translational Bioinformatics* 21.

De Leo, G., LeRouge, C., Ceriani, C., & Niederman, F. (2006). Websites most frequently used by physician for gathering medical information. *AMIA Annual Symposium Proceedings* 2006:902.

Ely, JW., Osheroff, JA., Chambliss, ML., Ebell, MH., and Rosenbaum, ME. (2005). Answering Physicians' Clinical Questions: Obstacles and Potential Solutions. *Journal of the American Medical Informatics Association* 12.2:217-224.

Ely, JW., Osheroff, JA., Gorman, PN., Ebell, MH., Chambliss, ML., and Pifer EA. (2000). A Taxonomy of Generic Clinical Questions: Classification Study. *British Medical Journal BMJ* 321:429–32.

Garla, VN., Brandt, C. (2013). Knowledge-based biomedical word sense disambiguation: an evaluation and application to clinical document classification. *Journal of the American Medical Informatics Association 2*0:882-6.

Johnson, S. B., & Gottfried, M. (1989, November). Sublanguage Analysis as a Basis for a Controlled Medical Vocabulary. *AMIA Annual Symposium Proceedings on Computer Applications in Medical Care* 519-523.

Jonnalagadda, S. R., Del Fiol, G., Medlin, R., Weir, C., Fiszman, M., Mostafa, J., & Liu, H. (2013). Automatically extracting sentences from Medline citations to support clinicians' information needs. *Journal of the American Medical Informatics Association*, *20*(5): 995-1000.

Lakiotaki, K., Hliaoutakis, A., Koutsos, S., & Petrakis, E. G. (2013, March). Towards personalized medical document classification by leveraging UMLS semantic network. In *International Conference on Health Information Science Springer Berlin Heidelberg* 93-104.

Lally, A., Prager, J. M., McCord, M. C., Boguraev, B. K., Patwardhan, S., Fan, J., Fodor, P. and Chu-Carroll, J. (2012). Question analysis: How Watson reads a clue. *IBM Journal of Research and Development 56*(3.4), 2-1.

Lang, FM. (2014, June 30). MetaMap 2012 Machine Output Explained. Retrieved from https://metamap.nlm.nih.gov/Docs/2012_MMO.pdf

Lindberg, D., Humphreys B., and McCray A. (1993). The Unified Medical Language System. *Methods of Information in Medicine* 32.4:281-291.

McCallum, AK. (2002) "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu.

Lucchiari, C., & Pravettoni, G. (2012). Cognitive balanced model: a conceptual scheme of diagnostic decision making. *Journal of evaluation in clinical practice*, *18*(1): 82-88.

Niu, Y., Hirst, G., McArthur, G., and Rodriguez-Gianolli, P. (2003). Answering clinical questions with role identification. *In Proceedings of 41st annual meeting of the*

Association for Computational Linguistics, Workshop on Natural Language Processing in Biomedicine 73–80.

Penn Treebank Project. Accessed at https://www.cis.upenn.edu/~treebank/.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D., Johnson, N., Cohen, K. B., & Duch, W. (2007). A shared task involving multi-label classification of clinical free text. *Proceedings of ACL BioNLP Prague, Czech Republic: Association for Computational Linguistics.* 97–104.

Pestian, J. P., Deleger, L., Savova, G. K., Dexheimer, J. W., & Solti, I. (2012). Natural Language Processing–The Basics. In *Pediatric Biomedical Informatics Springer Netherlands.* 149-172.

Sasaki, Y., Rea, B., & Ananiadou, S. (2007). Multi-topic aspects in clinical text classification. *BIBM 2007 Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine* 62-70.

Savova G., Masanz J., Ogren P., et al. (2010). Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc* 17:507–13.

Specialist Lexicon, a biomedical lexicon. Accessed at https://specialist.nlm.nih.gov/lexicon.

Srihari S. (2010). Machine Learning: Generative and Discriminative Models http://www.cedar.buffalo.edu/~srihari/CSE574/Discriminative-Generative.pdf

Suominen, H., Ginter, F., Pyysalo, S., Airola, A., Pahikkala, T., Salanter, S., and Salakoski, T. (2008). Machine learning to automate the assignment of diagnosis codes to

free-text radiology reports: a method description. *Proceedings of the ICML/UAI/COLT Workshop on Machine Learning for Health-Care Applications*.

Text Retrieval Conference (TREC) Home Page. Accessed at http://trec.nist.gov/.

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2:45-66.

Weka, machine learning package. Retrieved from http://www.cs.waikato.ac.nz/ml/weka/.

WordNet, an English lexical database. Retrieved from https://wordnet.princeton.edu/wordnet/.

APPENDIX A

A SAMPLE QUESTION IN XML IN THE ORIGINAL QUESTIONS DATA SET

```
<ROW>
    <nlm_id>NQ000975</nlm_id>
    <taxonomy_code>
    1421
</taxonomy_code>
    <short_question>
What&#39;s the name of the fracture when you pull off a piece of bone with the ligament?
</short_question>
    <general_question>
What is the name of the type of fracture where a piece of bone is pulled off with the ligament?
</general_question>
    <original_question>
What&#39;s the name of the fracture when you pull off a piece of bone with the ligament?
</original_question>
    <sub_question_present>No</sub_question_present>
    <generic_type>
    What is the name of that condition?
</generic_type>
    <local_source_field1>
    orthopedics
</local_source_field1>
    <purpose>To Deliver Healthcare</purpose>
    <content>Management</content>
    <disease_category>Musculoskeletal Diseases</disease_category>
    <keyword>Fractures</keyword>
    <annotations>NULL</annotations>
    <disease_category_list>Musculoskeletal Diseases</disease_category_list>
    <patient_type>Specific Patient</patient_type>
    <patient_age>Adolescence</patient_age>
    <patient_gender>Male</patient_gender>
    <trimester>NULL</trimester>
    <location>Office Visit</location>
    <visit_reason>
    Foot Injuries
</visit_reason>
    <notes>
    football injury to foot
</notes>
    <asker_occupation>Medicine</asker_occupation>
    <asker_specialty>Family Practice</asker_specialty>
    <professional_status>Licensed/Credentialed</professional_status>
    <asker_age> 33 </asker_age>
    <asker_gender>Female</asker_gender>
    <years_education>NULL</years_education>
    <year_asked>1997</year_asked>
    <date_received>December 19, 2001</date_received>
    <date_modified>December 19, 2001</date_modified>
    <editors_date>October 28, 2000</editors_date>
    <source>IOWA</source>
    <collection_id>IA</collection_id>
    <source_id>IA-0975</source_id>
    <citation>
        BMJ 1999;319:358-361
    </citation>
    <pubmed_id>
        10435959
    </pubmed_id>
    <local_source_field4>NULL</local_source_field4>
    <url>
        http://bmj.com/cgi/content/full/319/7206/358?view=full&pmid=10435959
    </url>
    <url_description>
        BMJ article entitled Analysis of questions asked by family doctors regarding patient care
    </url_description>
</ROW>
```

Figure 2: Sample Clinical Question in XML

APPENDIX B

TAXONOMY & CODES OF CLINICAL QUESTIONS

Table 5: Codes and Categories of Clinical Questions

CODE	PRIMARY	SECONDARY	TERTIARY	QUATERNARY	GENERIC TYPE	FREQUENCY (%)	COMMENTS
1.1.1.1	diagnosis	cause/ interpretation of clinical finding	symptom		What is the cause of symptom x? OR What is the differential diagnosis of symptom x? OR Could symptom x be condition y or be a result of condition y? OR What is the likelihood that symptom x is coming from condition y?	115 (8.2)	In 1.1.x.1, you start with a finding and you want to know what condition is causing it. You know what the finding is, you don't know what the condition is. See comment 1.2.1.1
1.1.2.1	diagnosis	cause/ interpretation of clinical finding	sign		What is the cause of physical finding x? OR What is the differential diagnosis of physical finding x? OR Could physical finding x be condition y or be a result of condition y? OR What is the likelihood that sign x is coming from condition y? OR At what level does physical finding x become clinically important? OR What is considered normal for physical finding x?	67 (4.8)	In 1.1.x.1, you start with a finding and you want to know what condition is causing it. You know what the finding is, you don't know what the condition is. See comment 1.2.1.1

1.1.3.1	diagnosis	cause/ interpretation of clinical finding	test finding (lab, ECG, imaging, biopsy, skin test, etc.)		What is the cause of test finding x? OR What is the differential diagnosis of test finding x? OR Could test finding x be condition y or be a result of condition y? OR What is the likelihood that test finding x is coming from condition y? OR How should I interpret test finding(s) x? OR How should I use test finding x in my decision? OR At what level does the value of test x become clinically important? OR What are the normal values (reference range) of test x?	64 (4.6)	In 1.1.x.1, you start with a finding and you want to know what condition is causing it. You know what the finding is, you don't know what the condition is. See comment 1.2.1.1
1.1.4.1	diagnosis	cause/ interpretation of clinical finding	unspecified findings or multiple categories of findings		Could this patient have condition y (given findings x1, x2, . . ., xn)? OR What is the likelihood that this patient has condition y (given findings x1, x2, . . ., xn)? OR What does this patient have (given findings x1, x2, . . ., xn)? OR What is the differential diagnosis of these findings?	51 (3.7)	In 1.1.x.1, you start with a finding and you want to know what condition is causing it. You know what the finding is, you don't know what the condition is. See comment 1.2.1.1

1.2.1.1	diagnosis	criteria/ manifestations			What are the manifestations (findings) of condition y? OR What is condition y? OR What does condition y look like? OR What are the criteria for diagnosis of condition y? OR How do I diagnose condition y (based on information I have or could get)? OR How do I distinguish between conditions y1, y2, …yn (based on information I have or could get)? OR How can you tell if the patient has condition y (based on information I have or could get)? OR Can condition y cause manifestation (finding) x? OR How does condition y cause manifestation (finding) x? OR Why did condition y cause manifestation (finding) x? OR Can condition y present with (as) manifestation (finding) x?	30 (2.2)	In 1.2.1.1, you start with a condition and you want to know if findings x1, x2, . . ,xn could be manifestations of that condition. You know what the condition is, you don't know if findings x1, x2, . . . , xn could be manifestations of that condition. See comment 1.1.x.1. The focus is on the condition, not the test: "How do I diagnose condition y?" could be either 1.3.1.1 or 1.2.1.1 depending on this focus. See comment 1.3.1.1.

1.3.1.1	diagnosis	test (lab, skin test, biopsy, imaging, element of physical exam, etc)	indications/ efficacy		Is test x indicated in situation y? OR What test (or evaluation, or work up), if any, is indicated/appropriate in situation y or with clinical findings x1, x2, . . , xn? OR What is the best test in situation y? OR Do the benefits of doing test x (work up x) outweigh the risks? OR How do I diagnose condition y (meaning what test(s) or work up should I do)? OR How do I distinguish between conditions y1, y2, …yn (meaning what test(s) or work up should I do)? OR How can you tell if the patient has condition y (meaning what test(s) or work up should I do)? OR Should this kind of patient have screening test x? OR What screening tests should this patient have?	112 (8.0)	The primary question is "What test should I do?" without regard to the quality/accuracy/performance characteristics of the test itself. The focus is on the indications for doing the test, not the characteristics of the test (see comment 1.3.2.1.) Also the focus is on the test, not the condition: "How do I diagnose condition y?" could be either 1.3.1.1 or 1.2.1.1 depending on the focus. See comment 1.2.1.1. Do not use this category for tests mandated by nonmedical organizations (5.2.1.1) and do not use it for drug levels (2.1.11.1)
1.3.2.1	diagnosis	test (lab, ECG, imaging, biopsy, skin test, element of physical exam, etc)	accuracy		How good is test x in situation y? OR What are the performance characteristics (sensitivity, specificity, etc.) of test x in situation y? OR What is the efficacy of screening with test x? OR What is the efficacy of screening for condition y?	14 (1.0)	The primary question is "How good is the test?" without regard to the indications for doing it. The focus here is on the characteristics of the test, not the indications for using it. See comment 1.3.1.1.

1.3.3.1	diagnosis	test (lab, ECG, imaging, biopsy, skin test, element of physical exam, etc)	timing/ monitoring		When (timing, not indications) should I do test x? OR When (timing, not indications) should I do test x to monitor condition y? OR When (timing, not indications) or how often should screening test x be done? OR When (timing, not indications) or how often should you screen for condition y?	31 (2.2)	Do not use for drug levels (2.1.11.1).
1.3.4.1	diagnosis	test (lab, ECG, imaging, biopsy, skin test, element of physical exam, etc)	preparation		What is the preparation for test x?	3 (0.2)	This category refers to what the patient must do before the test is performed. See comment 1.3.5.1.
1.3.5.1	diagnosis	test (lab, ECG, imaging, biopsy, skin test, element of physical exam, etc)	method		How do you do test x? OR What is the best way (best technique, best method) to do test x or screening test x?	6 (0.4)	This category refers to what the provider does during the performance of the test; how the test is done. See comment 1.3.4.1.
1.4.1.1	diagnosis	name finding	body part (anatomy) on physical exam or imaging study		What is the name of this body part? OR What is the anatomy here?	8 (0.6)	
1.4.2.1	diagnosis	name finding	condition		What is the name of that condition?	6 (0.4)	I know what the condition is, I just don't know its name. See comment 1.5.1.1
1.4.3.1	diagnosis	name finding	test		What is the name of that test?	2 (0.1)	I know what the test is, I just don't know its name. See comment 1.5.2.1

1.5.1.1	Diagnosis	orientation	condition		What is condition y?	5 (0.4)	I know the name of the condition, but I don't know what it is. See comment 1.4.2.1. This code will never be used in any analysis; all 1.5.1.1 codes will be converted to 1.2.1.1. It is included here only as an aid to the coder.
1.5.2.1	diagnosis	orientation	test		What is test x?	1 (0.1)	I know the name of the test, but I don't know what it is. See comment 1.4.3.1
1.6.1.1	diagnosis	inconsistencies			Why were this patient's findings (or course) inconsistent with usual expectations?	8 (0.6)	
1.7.1.1	diagnosis	cost			What is the cost of test x?	1 (0.1)	
1.8.1.1	diagnosis	not elsewhere classified			Generic type varies.	1 (0.1)	In a broad sense, the question is about diagnosis, but it does not fit any other diagnosis category.
2.1.1.1	treatment	drug prescribing	how to prescribe	undifferentiated	How do you prescribe/administer drug x (in situation y)?	10 (0.7)	
2.1.1.2	treatment	drug prescribing	how to prescribe	dosage	What is the dose of drug x (in situation y)? OR Should I change the dose of drug x (in situation y)? OR What is the maximum dose of drug x (in situation y)? OR What are equivalent doses among members of drug class x?	94 (6.7)	
2.1.1.3	treatment	drug prescribing	how to prescribe	timing	When (timing, not indication) or how should I start/stop drug x? OR How long should I give	27 (1.9)	Includes preventive drug treatment (and immunizations).

					drug x? OR When (timing, not indication) should I give drug x (in situation y)?		
2.1.2.1	treatment	drug prescribing	efficacy/ indications/ drug of choice	treatment	Is drug x (or drug class x) indicated in situation y or for condition y? OR What are the indications for drug x? OR Is any drug indicated for situation y? OR Does drug x work for condition y? OR How effective is drug x for condition y? OR What is the drug of choice for situation y or for condition y? OR What are the options for drug treatment of situation y or condition y? OR Is drug x1 better than drug x2, x3, . . ., xn for condition y? OR Is drug x1 just as effective as drug x2 (in situation y)? OR Does the benefit of giving drug x outweigh the risk?	150 (10.7)	Use 2.2.1.1 if treatments other than drugs could be considered. When the question does not specify drug treatment, the distinction between 2.2.1.1 (treatment in general) and 2.1.2.1 (drug treatment) can be difficult. The coder must judge whether nondrug treatment is a reasonable consideration.
2.1.2.2	treatment	drug prescribing	efficacy/ indications/ drug of choice	prevention	Should this kind of patient get prophylactic drug x to prevent condition y? OR Is prophylactic drug x indicated to prevent condition y? OR What prophylactic drug should I give to prevent condition y? OR How effective is prophylactic	40 (2.9)	Immunizations are drugs. Timing questions should be coded as 2.1.1.3.

					drug x in preventing condition y? OR For how long is drug x effective in preventing condition y? OR Is prophylactic drug x1 better than prophylactic drug x2 in preventing condition y?		
2.1.3.1	treatment	drug prescribing	adverse effects	findings caused by drug/ adverse effects of drug	Could finding y be caused by drug x? OR Does drug x cause finding y? OR What are the adverse effects of (or risks of using) drug x? OR What is the likelihood (incidence) of adverse effect(s) y resulting from drug x? OR How long do the adverse effects from drug x last after stopping it? OR Which drug has the fewest adverse effects? OR Are there differences among drugs x1, x2, . . ., xn in their likelihood of causing adverse effect(s) y?	59 (4.2)	
2.1.3.2	treatment	drug prescribing	adverse effects	administration in face of adverse effects	How can drug x be administered without causing adverse effect y or minimizing adverse effect y or in spite of adverse effect y? OR What dose of drug x would cause adverse effect y or any adverse	3 (0.2)	

					effect?		
2.1.3.3	treatment	drug prescribing	adverse effects	safety/ contraindications (includes pregnancy and breast feeding)	Is drug x safe to use in situation y? OR Is drug x contraindicated in situation y?	24 (1.7)	
2.1.4.1	treatment	drug prescribing	interactions		Is it OK to use drug x with drug y? OR Are there any interactions between drug x1 and drug (or food) x2, x3, . . . Xn?	28 (2.0)	
2.1.5.1	treatment	drug prescribing	name finding		What is the name of that drug?	12 (0.9)	I know what the drug is, I just don't know its name. See comment 2.1.6.1.

2.1.6.1	treatment	drug prescribing	orientation/ composition		What is drug x? OR What is in drug x (or dietary product x)? OR How much of component y is in drug x?	26 (1.9)	I know the name of the drug but I don't know what it is. See comment 2.1.5.1.
2.1.7.1	treatment	drug prescribing	physical characteristics		What are the physical characteristics (dosage forms, tablet/liquid characteristics, container characteristics) of drug x?	26 (1.9)	
2.1.8.1	treatment	drug prescribing	pharmacodynamics/ absorption		What are the pharmacodynamic/ absorption characteristics of drug x? OR How do the pharmacodynamic/ absorption characteristics of drugs x1, x2, . . ., xn compare?	2 (0.1)	
2.1.9.1	treatment	drug prescribing	mechanism of action		What is the mechanism of action of drug x? OR How does drug x work?	3 (0.2)	

2.1.10.1	treatment	drug prescribing	cost		What is the cost of drug x? OR How does the cost of drug x1 compare with the cost of drug x2, x3, . . . , xn?	11 (0.8)	
2.1.11.1	treatment	drug prescribing	serum levels		What are the indications for getting a drug serum level or what time should it be drawn or how often should it be drawn?	1 (0.1)	
2.1.12.1	treatment	drug prescribing	availability		Is drug x available yet? OR Is drug x available over-the-counter?	4 (0.3)	
2.2.1.1	treatment	not limited to but may include drug prescribing	efficacy/ indications	treatment	How should I treat finding/condition y (given situation z)? OR Should I use treatment/procedure x for condition/finding y? OR What is the efficacy of treatment/procedure x (for condition y)? OR Does	82 (5.9)	When the question does not specify drug treatment, the distinction between 2.2.1.1 (treatment in general) and 2.1.2.1 (drug treatment) can be difficult. The coder must judge whether nondrug treatment is a reasonable consideration.

					procedure/treatment x work (for condition y)? OR Is treatment/procedure x indicated (for condition y)? OR What is the best treatment/procedure to do (for condition y)? OR Does the benefit of treatment/procedure x outweigh the risk? OR What are the options for treatment of condition y (in situation z)? OR Is there any treatment for condition y? OR What is the goal of treatment of condition y? OR At what level of severity of condition y is treatment indicated?		
2.2.1.2	treatment	not limited to but may include drug prescribing	efficacy/indications	prevention	Should this kind of patient get prophylactic treatment (intervention) x to prevent condition y? OR Is prophylactic treatment (intervention) x indicated to prevent condition y? OR What prophylactic treatment (intervention) should I give to prevent condition y? OR Does treating condition y1 help prevent condition y2?	1 (0.1)	

2.2.2.1	treatment	not limited to but may include drug prescribing	timing		When (or how) should I start/stop treatment x? OR When (timing, not indication) should I use treatment x (in situation y)? OR How long should I continue treatment x for condition y?	2 (0.1)	
2.2.3.1	treatment	not limited to but may include drug prescribing	how to do it		How do you do treatment/procedure x? OR What is the best way to do treatment/procedure x?	1 (0.1)	Do not use for diagnostic methods (1.3.5.1).
2.2.4.1	treatment	not limited to but may include drug prescribing	principles/ rationale		What are the principles (or rationale) behind therapy x? OR How does therapy x work?	1 (0.1)	
2.3.1.1	treatment	not elsewhere classified			Generic type varies	4 (0.3)	In a broad sense, the question is about treatment, but it does not fit any other treatment category.

3.1.1.1	management (not specifying diagnostic or therapeutic)	condition/ finding			How should I manage condition/finding/situation y? (not specifying diagnostic or therapeutic management) OR What management options are there in situation y? OR How aggressive/conservative should I be in situation y?	67 (4.8)	Do not use this code for questions about only diagnosis (1.3.1.1) or only treatment (2.2.1.1). Do not use this code if you know the diagnosis. Go with the meaning, not with the words: If the questioner says "management," but treatment is the only reasonable kind of management (the diagnosis is not in question), do not use this code.
3.2.1.1	management (not specifying diagnostic or therapeutic)	other providers	practices of other providers		Why did provider x treat the patient this way? OR How do other providers manage condition y?	12 (0.9)	
3.2.2.1	management (not specifying diagnostic or therapeutic)	other providers	referral		When should you refer in situation y?	8 (0.6)	

3.2.3.1	management (not specifying diagnostic or therapeutic)	other providers	community services		What social services (or support groups, community groups) are available for condition/situation y?	5 (0.4)	
3.3.1.1	management (not specifying diagnostic or therapeutic)	doctor-patient communication	how to advise		How should I advise the patient/family in situation y?	8 (0.6)	
3.3.2.1	management (not specifying diagnostic or therapeutic)	doctor-patient communication	how to approach difficult issue		What is the best way to discuss or approach discussion of difficult issue x?	5 (0.4)	
3.3.3.1	management (not specifying diagnostic or therapeutic)	doctor-patient communication	patient compliance		How can I get the patient/family to comply with my recommendations or agree with my assessment?	21 (1.5)	

3.4.1.1	management (not specifying diagnostic or therapeutic)	not elsewhere classified			Generic type varies	0 (0.0)	In a broad sense, the question is about management, but it does not fit any other management category.
4.1.1.1	epidemiology	prevalence/ incidence			What is the incidence/prevalence of condition y (in situation z)? OR Why is the incidence/prevalence of condition y changing?	14 (1.0)	This category is plain incidence or plain prevalence. I am not interested in associations between risk factors and conditions. I am not interested in associations among different conditions. See comments 4.2.1.1 and 4.3.1.1.
4.2.1.1	epidemiology	etiology	causation/ association	risk factors/ disease agents	Is x a risk factor for condition y? OR Is x associated with condition y? OR Is condition y1 associated with condition y2, y3, . . . yn (all conditions present at the same time)? OR Can finding or disease-agent x cause condition y? OR What are the causes of condition y? OR What conditions or risk factors are associated with condition y? OR Why did the patient get condition y?	40 (2.9)	This category asks about associations between a risk factor and a condition (the risk factor occuring before the condition) or between 2 or more conditions (that are present at the same time). Do not use this code for 2 elements that are part of the same disease process (which is 4.3.1.1). Instead, use it when one element, which is not part of the disease process, is a risk factor for the condition (disease). See comments 4.1.1.1 and 4.3.1.1. Do not use this category for adverse drug reactions (2.1.3.1).

4.2.1.2	Epidemiology	etiology	causation/ association	genetics	Is condition y hereditary?	2 (0.1)	
4.3.1.1	epidemiology	course/ prognosis			What is the usual course (or natural history) of condition y? OR What is the prognosis (or likelihood of complications) of condition/situation y? OR Can condition y1 lead to condition y2, y3, . . . yn (condition y1 occurs before conditions y2, y3, . . ., yn)?	25 (1.8)	This category asks what happens to a patient over time. It includes plain prognosis questions as well as associations between 2 conditions, where one condition occurs before the other. See comments 4.1.1.1 and 4.2.1.1).
4.4.1.1	epidemiology	not elsewhere classified			Generic type varies.	1 (0.1)	In a broad sense, the question is about epidemiology, but it does not fit any other epidemiology category.

5.1.1.1	nonclinical	education	provider	continuing medical education	I need to learn more about topic x. OR I need to review topic x.	8 (0.6)	
5.1.1.2	nonclinical	education	provider	information source	Where can I find or how can I get information about topic x? OR Is there any information on topic x?	5 (0.4)	
5.1.1.3	nonclinical	education	provider	trainee	How can I better teach this trainee (medical student, resident, other provider)?	1 (0.1)	
5.1.2.1	nonclinical	education	patient		What patient education materials are available for situation y? OR Where can I get patient education materials on topic x?	3 (0.2)	

5.2.1.1	nonclinical	administration			What are the administrative rules/ considerations in situation y? OR What are the local requirements and issues relevant to situation y? OR What are the safety issues for health care workers in situation y?	13 (0.9)	Examples: disease codes, procedure codes, HMO rules, insurance company rules, employer rules, government rules. Distinguish between guidelines primarily based on clinical issues where the patient's welfare is the primary concern (consider other codes) versus guidelines designed to meet the goals of the organization.
5.3.1.1	nonclinical	ethics			What are the ethical considerations in situation y?	4 (0.3)	
5.4.1.1	nonclinical	legal			What are the legal considerations in situation y?	1 (0.1)	

5.5.1.1	nonclinical	frustration			Generic type varies. Not a true question, but rather an expression of frustration or an unanswerable dilemma.	15 (1.1)	
5.6.1.1	nonclinical	not elsewhere classified			In a broad sense, the question is nonclinical, but it does not fit any other nonclinical category.	2 (0.1)	
6.1.1.1	unclassified				Generic type varies. Unable to classify	0 (0.0)	